USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA
FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

Una introducción a los modelos aditivos generalizados

Mariña García Ponte

2023/2024

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRADO DE MATEMÁTICAS

Traballo Fin de Grado

Una introducción a los modelos aditivos generalizados

Mariña García Ponte

Julio, 2024

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Trabajo propuesto

Área de Conocimiento: Estadística e Investigación Operativa.
Título: Una introducción a los modelos aditivos generalizados.
Breve descripción del contenido
En la actualidad, debido a la gran complejidad de los datos con los que se trabaja, a menudo algunas de las hipótesis de los modelos clásicos de regresión (normalidad de los errores, linealidad en la relación entre la respuesta y las variables explicativas) resultan muy restrictivas. Surgen de esta forma los modelos aditivos generalizados (GAM, del inglés "generalized additive models"). Estos modelos, además de relajar la hipótesis de normalidad de los errores, permiten una mayor flexibilidad en la relaciones entre las variables, introduciendo, para ello, funciones *suaves*. En este trabajo haremos una introducción de los modelos de regresión más sencillos y exploraremos sus limitaciones, justificando de esta manera el planteamiento del modelo aditivo generalizado. Abordaremos los principales métodos de estimación de las componentes *suaves* mediante splines y los ilustraremos utilizando un ejemplo de datos reales.
Recomendaciones
Conocimiento básico del software estadístico  .

Índice

Resumen viii

Preámbulo xi

1. Modelos de regresión lineales: revisión y limitaciones **1**

1.1. Modelos de regresión clásicos 1

1.1.1. Modelo de regresión lineal simple 1

1.1.2. Modelo de regresión lineal múltiple 2

1.1.3. Modelo lineal general 3

1.2. Modelo lineal generalizado 8

1.2.1. Estimación de los parámetros del modelo 10

1.2.2. Medidas de bondad de ajuste 15

1.2.3. Modelo logístico 17

1.2.4. Modelo de Poisson 18

1.2.5. Limitaciones del modelo lineal generalizado 18

1.3. Introducción a los modelos aditivos 20

2. Modelos aditivos generalizados: formulación e ideas básicas **23**

2.1. El concepto de spline 23

2.2. Splines de suavización 25

2.3. Splines de regresión 26

2.4. Regresión spline penalizada 30
2.5. Elección del parámetro de suavizado 33
2.6. Regresión spline penalizada para respuesta generalizada 34
2.7. Ejemplo ilustrativo con datos simulados 37

3. Aplicación a datos reales **43**

3.1. Descripción de la base de datos 43
3.2. Análisis exploratorio de los datos 46
3.3. Ajuste de los modelos 47
3.4. Conclusiones 54

I. Código de R **55**

Resumen

Los modelos aditivos generalizados representan una herramienta muy útil en el análisis de datos debido a su flexibilidad y capacidad para modelar relaciones no lineales entre variables. En este trabajo, se llevará a cabo una revisión de los modelos de regresión lineales y lineales generalizados, exponiendo sus limitaciones y la necesidad de emplear métodos más flexibles, como los modelos aditivos generalizados. Estos modelos introducen funciones *suaves* para modelar las relaciones entre la variable respuesta y las variables explicativas. Se presentará su formulación teórica y se examinarán los principales métodos de estimación mediante splines. Los modelos introducidos, así como sus limitaciones, serán ilustrados a través de simulaciones. Finalmente, se presentará una aplicación del modelo aditivo generalizado a una base de datos reales. Este ejemplo permitirá ilustrar sus ventajas en un contexto real, donde la capacidad de adaptación a patrones no lineales es esencial para obtener resultados precisos y útiles.

Abstract

Generalized additive models represent a very useful tool in data analysis due to their flexibility and ability to model non-linear relationships between variables. In this work, a review of linear and generalized linear regression models will be conducted, exposing their limitations and the need to employ more flexible methods, such as generalized additive models. These models introduce *smooth* functions to model the relationships between the response variable and the explanatory variables. Their theoretical formulation will be presented, and the main estimation methods using splines will be examined. The introduced models, along with their limitations, will be illustrated through simulations. Finally, an application of the generalized additive model to a real database will be presented. This example will illustrate the advantages of this model in a real context, where the ability to adapt to non-linear patterns is essential for obtaining accurate and useful results.

Introducción

Hoy en día, la creciente complejidad de los datos que se manejan en diversas disciplinas ha impulsado la necesidad de plantear modelos de regresión cada vez más flexibles. Los modelos lineales clásicos, a pesar de su fácil interpretación e implementación, a menudo resultan insuficientes para explicar de forma adecuada y precisa las relaciones en ciertos conjuntos de datos. En este contexto, los modelos aditivos generalizados aparecen como una herramienta de gran utilidad. Estos modelos, a diferencia de los modelos de regresión clásicos, permiten que la variable respuesta siga cualquier distribución de la familia exponencial e incorporan funciones *suaves* en el modelado de las relaciones entre la variable respuesta y las variables explicativas, relajando así la hipótesis de linealidad. Surge, de esta forma, el problema de estimar estas funciones *suaves* de forma que no se produzca un sobreajuste de los datos ni se pierda la capacidad de capturar las relaciones existentes entre las variables.

El principal objetivo de este trabajo es realizar una introducción a los modelos aditivos generalizados. Para ello, justificaremos su planteamiento exponiendo las limitaciones de los modelos lineales y abordaremos las principales técnicas de estimación de las componentes *suaves* del modelo mediante splines. Además, ilustraremos estos resultados mediante simulaciones y presentaremos un ejemplo de aplicación a datos reales. El contenido de este trabajo se organiza de la manera descrita a continuación.

En el capítulo uno, introduciremos los modelos de regresión clásicos y, en el contexto del modelo de regresión general, abordaremos brevemente las cuestiones estimación e inferencia sobre los parámetros. Ilustraremos las limitaciones de estos modelos a través de la simulación de datos y se discutirá cómo estas limitaciones pueden afectar la precisión e interpretación de los resultados. A continuación, generalizaremos estos modelos, presentando el modelo lineal generalizado. Por su similitud con la estimación de los modelos aditivos generalizados, profundizaremos en la estimación de sus parámetros y definiremos medidas de bondad de ajuste. Como casos particulares de este tipo de modelos consideraremos el modelo logístico y el modelo de Poisson. Por último, exploraremos las limitaciones de este tipo de modelos y la necesidad de contar con modelos más flexibles, como el modelo aditivo y el modelo aditivo generalizado.

En el capítulo dos, partiremos de un modelo de respuesta normal en el que la relación entre la variable respuesta y la variable explicativa viene dada por una función *suave* de la que desconocemos su forma. En primer lugar, abordaremos el concepto de spline y su utilidad para estimar estas funciones *suaves*, destacando las principales similitudes y diferencias con la estimación de los modelos del capítulo uno. Explicaremos distintos métodos de estimación mediante splines, como los splines de suavización, los splines de regresión y los splines de regresión penalizada. Estudiaremos la estimación de estas componentes *suaves* en un modelo aditivo con respuesta generalizada y varias variables explicativas y lo ilustraremos haciendo uso de la simulación de datos. Además, también compararemos las ventajas de los modelos aditivos generalizados frente a modelos estudiados en el capítulo uno.

El capítulo tres consiste en un estudio de la base de datos disponible en Wolberg et al. (1995) sobre el cáncer de pecho. En primer lugar, describiremos brevemente el contenido de los datos, definiendo las variables que vamos a considerar. A continuación, realizaremos un primer análisis exploratorio de las variables para entender su distribución y posibles relaciones entre ellas. Finalmente, procederemos con el ajuste de los modelos aditivos generalizados, partiendo de un modelo muy sencillo y aumentando la complejidad hasta conseguir un buen ajuste.

Las principales referencias utilizadas en este trabajo son Wood (2017), Faraway (2004) y Perperoglou et al. (2019). Para las simulaciones de datos, ajuste de los modelos y figuras de este trabajo se ha utilizado el software estadístico R. El código para reproducir los resultados de los capítulos uno y dos se recoge en el Anexo I. En él, se ha utilizado el paquete `mgcv` [1] de R para ajustar los modelos aditivos generalizados mediante estimación por splines.

[1] https://cran.r-project.org/web/packages/mgcv/index.html

Capítulo 1

Modelos de regresión lineales: revisión y limitaciones

En este capítulo haremos un breve recorrido a lo largo de los modelos de regresión clásicos, el modelo lineal generalizado y los modelos aditivos. Estos modelos serán la base para la construcción del modelo lineal generalizado que trataremos en el siguiente capítulo. Además, usaremos el software estadístico  para ilustrar las limitaciones de estos modelos mediante la simulación de datos y la necesidad de contar con modelos más flexibles, como lo son los modelos lineales generalizados.

1.1. Modelos de regresión clásicos

En esta sección revisaremos los modelos de regresión clásicos. Comenzaremos introduciendo el modelo más sencillo, el modelo lineal simple, que considera una única variable explicativa. A continuación, plantearemos el modelo lineal múltiple, que surge al considerar más de una variable explicativa. Por último, formularemos el modelo lineal general, que permitirá construir modelos con variables explicativas discretas y los conocidos como modelos linealizables. En el contexto del modelo lineal general, expondremos los principales resultados para la estimación de los parámetros e inferencia. Estos resultados se pueden consultar en Faraway (2004).

1.1.1. Modelo de regresión lineal simple

Los modelos de regresión sirven para representar la relación entre una variable respuesta Y y una o más variables explicativas.

El modelo de regresión más sencillo que se puede formular es el modelo lineal simple, que incluye una única variable explicativa numérica que denotaremos por X. Para su construcción, se considera una función de regresión, $f(x)$, que representa la media de la variable respuesta Y condicionada al valor de la variable explicativa X. Es decir, $f(x) = \mathbb{E}(Y \mid X = x)$. Se supone que $f(x)$ es una función lineal de la forma $f(x) = \beta_0 + \beta_1 x$, donde β_0 y β_1 son parámetros desconocidos.

Consideraremos que estamos trabajando bajo diseño fijo, es decir, que disponemos de una muestra de n individuos donde los valores de la variable explicativa están fijados. Denotaremos la muestra obtenida bajo diseño fijo como:

$$(x_1, Y_1), ..., (x_n, Y_n).$$

Por tanto, podemos plantear el modelo de regresión lineal simple como:

$$Y_i = f(x_i) + \varepsilon_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad i \in \{1, ..., n\} \tag{1.1}$$

donde ε_i son los errores del modelo. Para hacer inferencia sobre el modelo se supone que ε_1,...,$\varepsilon_n \in N(0, \sigma^2)$ y que además son independientes.

1.1.2. Modelo de regresión lineal múltiple

En un modelo de regresión lineal se puede incluir más de una variable explicativa numérica. Estos modelos se conocen como modelos de regresión lineal múltiple. Se considera una variable respuesta Y y $p-1$ variables explicativas, X_1,...,X_{p-1}. Bajo diseño fijo, el modelo se puede expresar como:

$$Y_i = \beta_0 + \beta_1 x_{i,1} + ... + \beta_{p-1} x_{i,p-1} + \varepsilon_i, \quad i \in \{1, ..., n\}$$

donde $x_{i,1}$, ..., $x_{i,p-1}$ son las variables explicativas asociadas al i-ésimo individuo; Y_i su variable respuesta y ε_i su error. Para hacer inferencia sobre el modelo, se supone que los errores cumplen que $\varepsilon_1, ..., \varepsilon_n \in N(0, \sigma^2)$ y son independientes entre sí.

Este modelo se puede escribir en notación matricial como:

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p-1} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np-1} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Es decir, $Y = X\beta + \varepsilon$, donde Y es el vector de respuestas, β el vector de parámetros y ε el vector de errores, que cumple $\varepsilon \in N_n(0, \sigma^2 I_n)$. X es una matriz $n \times p$ que se denomina matriz

de diseño. En el caso del modelo lineal múltiple, esta matriz contiene en cada columna las n observaciones de una de las $p-1$ características de los individuos (salvo en la primera columna, que está formada por unos, para incluir al intercepto).

1.1.3. Modelo lineal general

El modelo lineal múltiple se puede generalizar dando lugar al modelo lineal general. La suposición principal para la formulación de este modelo es que la variable respuesta Y se puede expresar como función lineal de los parámetros. Es decir,

$$Y = X\beta + \varepsilon$$

donde Y es el vector columna de dimensión n formado por las variables respuestas, X es una matriz no aleatoria de tamaño $n \times p$, β es un vector columna de dimensión p que contiene los parámetros del modelo y ε es un vector n-dimensional formado por los errores correspondientes a cada observación. Para llevar a cabo la estimación e inferencia del modelo se suponen ciertas cuatro hipótesis. La primera de ellas es la linealidad del modelo: la variable respuesta depende linealmente de las variables explicativas. El resto de las hipótesis son sobre los errores del modelo, que se suponen homocedásticos (la varianza no depende de la variable explicativa y es la misma para todos los individuos), independientes y normales. Es decir, $\varepsilon \in N_n(0, \sigma^2 I_n)$. Bajo esta formulación, la matriz X no tiene por qué contener los valores de las variables explicativas.

En el marco del modelo lineal general podemos incluir, por ejemplo, modelos linealizables, como son los modelos con interacciones o los modelos polinómicos. Además, también abarca modelos con variables explicativas categóricas, como los modelos de análisis de la varianza y análisis de la covarianza.

Modelos linealizables

Dentro del modelo lineal general se pueden incluir modelos en los que las columnas de la matriz de diseño se obtengan operando con las variables explicativas iniciales. Dos ejemplos de este tipo de modelos son los modelos de regresión polinómica y los modelos con interacciones.

El modelo de regresión polinómica se puede formular como:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + ... + \beta_k X^k + \varepsilon \tag{1.2}$$

donde Y es la variable respuesta; X es la variable explicativa unidimensional; $\beta_0, \beta_1, ..., \beta_k$ son los parámetros del modelo y ε es el error, que cumple $\varepsilon \in N_n(0, \sigma^2 I_n)$. Obsérvese que la variable respuesta sigue siendo una función lineal de los parámetros β_l, $l \in \{0, 1, \ldots, k\}$. Bajo

esta formulación, se están tratando las potencias de la variable explicativa original como nuevas variables.

En el marco del modelo lineal general es posible introducir interacciones entre las variables del modelo. Cuando las variables son continuas, la forma mas sencilla de incluir una interacción consiste en añadir una nueva variable definida como el producto de dos variables. Por ejemplo, en el caso de un modelo con variables X_1 y X_2, la interacción entre ambas se incluye añadiendo la variable X_1X_2:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2 + \varepsilon.$$

La variable respuesta Y es lineal respecto a los parámetros β_0, β_1, β_2 y β_{12}. A este último coeficiente se le llama coeficiente de interacción.

Variables explicativas discretas: modelo ANOVA y ANCOVA

Dentro del modelo lineal general es posible incluir variables explicativas discretas, dando lugar al modelo de análisis de la varianza (ANOVA) y al modelo de análisis de la covarianza (ANCOVA). En este trabajo consideraremos únicamente covariables continuas, pero se introducen estos modelos por completitud y ya que muchas de las herramientas que utilizaremos son fácilmente adaptables a modelos con covariables categóricas.

El modelo de análisis de la varianza considera una única variable explicativa discreta mientras que la respuesta Y sigue siendo una variable continua. Dada una muestra de n observaciones, se agrupan teniendo en cuenta el valor de la covariable: si esta toma I valores distintos, se considera la existencia de I grupos. En todos los grupos la varianza es la misma, mientras que la media puede variar entre grupos. Esto da lugar a I muestras independientes:

$$Y_{11}, Y_{12}, \ldots, Y_{1n_1} \text{ de una población } N(\mu_1, \sigma^2)$$

$$\vdots$$

$$Y_{I1}, Y_{I2}, \ldots, Y_{In_I} \text{ de una población } N(\mu_I, \sigma^2).$$

El modelo se puede formular como:

$$Y_{ij} = \mu_i + \varepsilon_{ij}, \quad \text{para } i \in \{1, \ldots, I\},\ j \in \{1, \ldots, n_i\}$$

donde los errores son independientes y cumplen $\varepsilon_{ij} \in N(0, \sigma^2)$ y μ_i es la media de cada población. Se trata de un modelo lineal general donde la matriz de diseño contiene unas variables conocidas como variables *dummy* que señalan si un individuo pertenece o no a cada uno de los grupos. La estimación de los parámetros del modelo μ_i se lleva a cabo mediante la media muestral de las observaciones de la variable respuesta correspondientes a cada una de las categorías.

Por otra parte, el modelo de análisis de la covarianza permite la introducción de variables discretas y continuas en un mismo modelo. Se puede formular como:

$$Y_{ij} = \mu + \alpha_i + \gamma z_{ij} + \varepsilon_{ij} \quad \text{para } i \in \{1, \dots, I\},\ j \in \{1, \dots, n_i\}.$$

De esta manera, se considera una constante μ a la que se le suma un efecto α_i, que depende del grupo i al que pertenezca el individuo Y_{ij}. Por tanto, este coeficiente α_i recoge el efecto de la variable categórica de forma similar al modelo ANOVA. La variable continua se incluye mediante la variable z_{ij} con coeficiente de regresión γ. Los errores $\varepsilon_{ij} \in N(0, \sigma^2)$ y son independientes entre sí. De esta forma, se está ajustando una recta distinta para cada uno de los I grupos. Todas ellas son paralelas, ya que tienen la misma pendiente, pero el intercepto varía en cada una de las poblaciones. Es posible flexibilizar este modelo mediante la introducción de interacciones entre la covariable discreta y continua. En ese caso, se ajustarían I rectas de regresión simple de manera independiente, es decir, permitiendo variar la pendiente y el intercepto en cada uno de los grupos.

Estimación de los parámetros del modelo

Los resultados que se verán a continuación son válidos en el marco del modelo lineal general. Los parámetros del modelo son: el vector β y la varianza del error σ, que se denomina parámetro *nuisance*. La estimación del vector de parámetros β se lleva a cabo mediante el método de mínimos cuadrados, que consiste en buscar un estimador de β, $\hat{\beta}$, que minimice la suma de los residuos al cuadrado:

$$\hat{\beta} = \min_{\beta} \sum_{i=1}^{n} (Y_i - x_i\beta)^2$$

donde x_i es la fila i-ésima de la matriz X. La solución de este problema de minimización se puede obtener de manera analítica y es el estimador de β,

$$\hat{\beta} = (X'X)^{-1}X'Y.$$

Obsérvese que las predicciones del modelo, $\hat{Y}$, se pueden obtener como:

$$\hat{Y} = X\hat{\beta} = X(X'X)^{-1}X'Y = HY$$

donde $H = X(X'X)^{-1}X'$ es una matriz cuadrada de orden n, simétrica e idempotente que se conoce como matriz hat. Por otra parte, el estimador de la varianza del error, σ^2, es:

$$\hat{\sigma}^2 = \frac{1}{n-p} \sum_{i=1}^{n} (Y_i - x_i\hat{\beta})^2 = \frac{\text{RSS}}{n-p} \tag{1.3}$$

donde RSS representa la suma de los residuos al cuadrado.

Propiedades de los estimador $\hat{\beta}$

A continuación expondremos las propiedades fundamentales de $\hat{\beta}$, que son la base para los procedimientos de inferencia sobre este estimador. Estos resultados son relevantes para estudiar el comportamiento de β y consecuentemente la relación entre las variables explicativas y la variable respuesta.

$\hat{\beta}$ es un estimador insesgado de β, es decir, $\mathbb{E}[\hat{\beta}] = \beta$. Esta propiedad se obtiene inmediatamente teniendo en cuenta que los errores tienen media 0. Por otra parte, gracias a la homocedasticidad de los errores, se puede demostrar que $\mathrm{Cov}(\hat{\beta}, \hat{\beta}) = \sigma^2(X'X)^{-1}$.

Dado que $Y \in N_n(X\beta, \sigma^2 I)$ y $\hat{\beta}$ no es más que una transformación lineal de Y, se deduce que $\hat{\beta}$ sigue una distribución normal de media y varianza calculadas antes:

$$\hat{\beta} \in N_p(\beta, \sigma^2(X'X)^{-1}).$$

Suponiendo que σ^2 es desconocida, se puede usar como pivote para los resultados de inferencia sobre β la siguiente variable:

$$\frac{(\hat{\beta} - \beta)'(X'X)(\hat{\beta} - \beta)}{p\ \hat{\sigma}^2} \in F_{p,n-p}.$$

Para la inferencia sobre un único coeficiente β_l se usa el pivote:

$$\frac{\hat{\beta}_l - \beta_l}{\hat{\sigma}\sqrt{(X'X)^{-1}_{ll}}} \in T_{n-p}, \qquad l = 0, ..., p-1.$$

Bondad de ajuste

Para evaluar la calidad del ajuste de un modelo, se utilizan medidas conocidas como medidas de bondad de ajuste. Estas medidas permiten cuantificar la discrepancia entre las observaciones reales y los valores ajustados por el modelo. Una de las más relevantes es el coeficiente de determinación R^2, que se calcula como:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

y refleja la proporción de la variabilidad en los datos que es explicada por el modelo. Por lo tanto, el cociente anterior representa la variabilidad que no es explicada por el modelo. En el numerador, aparece la suma de residuos al cuadrado, mientras que en denominador se encuentra la suma al cuadrado de las diferencias de los valores observados y el valor medio de las observaciones, $\bar{y}$. Este valor oscila entre cero y uno: un valor cercano a uno indica un buen ajuste, significando que

una gran parte de la variabilidad de los datos es explicada por el modelo. Por el contrario, un valor cercano a cero sugiere que el modelo explica muy poca variabilidad de los datos, indicando que no se ajusta bien a las observaciones.

Limitaciones del modelo lineal general

El modelo lineal general, a pesar de permitir la introducción de variables construidas a partir de transformaciones de las variables originales, resulta insuficiente para ajustarse adecuadamente a los datos en algunas ocasiones. A continuación ilustraremos mediante la simulación de datos dos de sus principales limitaciones, que vienen dadas por la suposición de las hipótesis de linealidad y normalidad y de los errores.

En el primer ejemplo simulamos unos datos donde la respuesta Y no sigue una distribución normal. En esta simulación, consideramos una variable respuesta dicotómica con distribución de Bernoulli, es decir, únicamente toma valores cero y uno. La variable explicativa toma 300 valores uniformemente distribuidos en el intervalo [-2,2]. Al ajustar un modelo lineal simple de la forma (1.1) para estos datos, como se puede ver en azul en la Figura 1.1, el modelo no respeta la naturaleza binaria de la respuesta. La recta de regresión toma valores distintos de cero y uno, e incluso fuera del intervalo [0,1] para valores muy pequeños o muy grandes de la variable explicativa. La curva verde representa el ajuste de un modelo polinómico de grado tres de la forma (1.2) y la curva roja el modelo polinómico de grado diez. Como se puede observar, por mucho que aumentemos la flexibilidad de la curva ajustada aumentando el grado del polinomio, ninguno de los modelos tiene en cuenta la distribución dicotómica de la variable respuesta.

En el segundo ejemplo generamos unos datos simulados de un modelo de la siguiente forma:

$$Y = f(x) + \varepsilon = 0.5 + \cos(2x) + e^{\cos(x)} + \varepsilon, \quad \varepsilon \in N(0, 0.5). \tag{1.4}$$

Tomamos como variable explicativa una secuencia de valores equiespaciados una distancia de 0.01 en el intervalo [-4,4]. Estos datos cumplen las hipótesis de normalidad, homocedasticidad e independencia de los errores. Sin embargo, como podemos ver en el diagrama de dispersión de la Figura 1.2, la dependencia entre la variable X y la respuesta Y es claramente no lineal.

Si intentamos ajustar estos datos con un modelo lineal de la forma (1.1), como aparece en azul en la Figura 1.2, la recta es incapaz de reflejar ninguna de las zonas de crecimiento ni decrecimiento que parece que sigue la nube de puntos. Esto se ve reflejado en el coeficiente de determinación, el cual es del orden de 10^{-6}.

Para permitir más flexibilidad a la curva, consideramos un modelo de regresión polinómico de grado tres de la forma (1.2). Como se puede ver en verde en la misma figura, este modelo

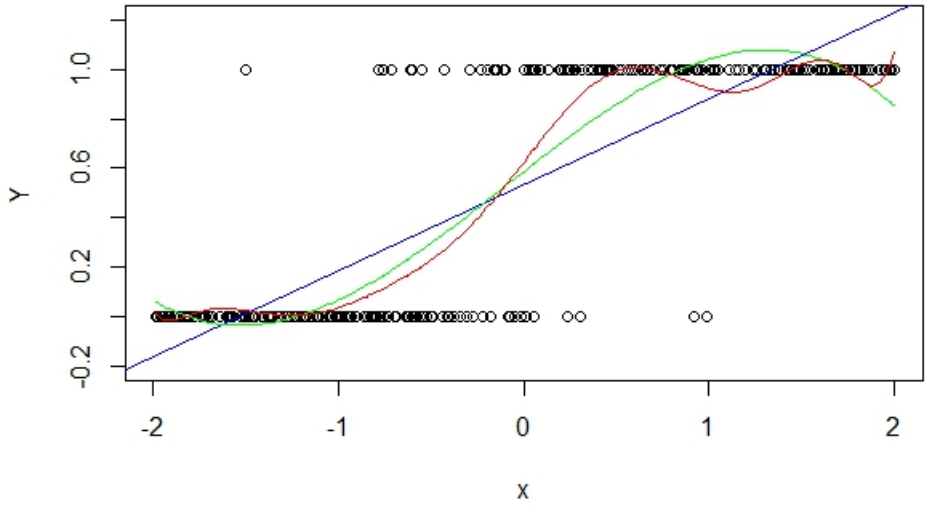


Figura 1.1: Diagrama de dispersión de la variable binaria Y. En azul, la recta de regresión lineal simple; en verde, el modelo polinómico de orden tres y en rojo el modelo polinómico de orden diez.

refleja la tendencia global del diagrama de dispersión, pero sigue siendo un modelo bastante inadecuado, con un coeficiente de determinación de 0.1077. Al ir incrementando el grado de los polinomios, observamos como los modelos se ajustan más a la nube de puntos y el coeficiente de determinación es cada vez más alto. Por ejemplo, la curva amarilla de la Figura 1.2 se trata de un modelo de regresión polinómico de orden 8. A simple vista podemos apreciar como es capaz de reflejar muy bien el patrón que sigue la nube de puntos. Además, consta de un coeficiente de determinación de 0.8059, el cual es considerablemente alto. Inicialmente podríamos pensar que se trata de un buen modelo. Sin embargo, en la práctica, es un modelo demasiado complejo, pues consta de nueve parámetros que determinar. Realizando operaciones sobre la variable explicativa original se podrían ajustar multitud de modelos linealizables distintos que podrían explicar mejor los datos siendo más sencillos. Sin embargo, el desconocimiento de la forma de la función f hace que sea más eficiente recurrir a otros métodos para su estimación, como veremos en el siguiente capítulo.

1.2. Modelo lineal generalizado

El modelo lineal generalizado surge al relajar la hipótesis de normalidad de la variable respuesta del modelo lineal general. Los resultados relativos al planteamiento y estimación de los

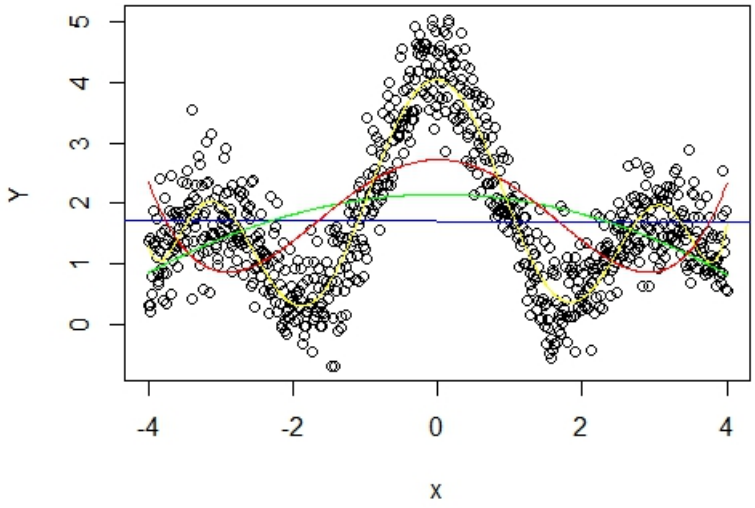


Figura 1.2: Diagrama de dispersión del modelo (1.4). En azul, la recta de regresión lineal simple; en verde, el modelo polinómico de grado tres; en rojo, el modelo polinómico de grado cuatro y en amarillo, el modelo polinómico de grado ocho.

parámetros de este modelo se pueden consultar en Wood (2017). Este modelo supone que las observaciones Y_i de la variable respuesta siguen cualquier distribución de la familia exponencial y son independientes entre sí. La función de densidad o probabilidad asociada a una variable con distribución perteneciente a la familia exponencial es:

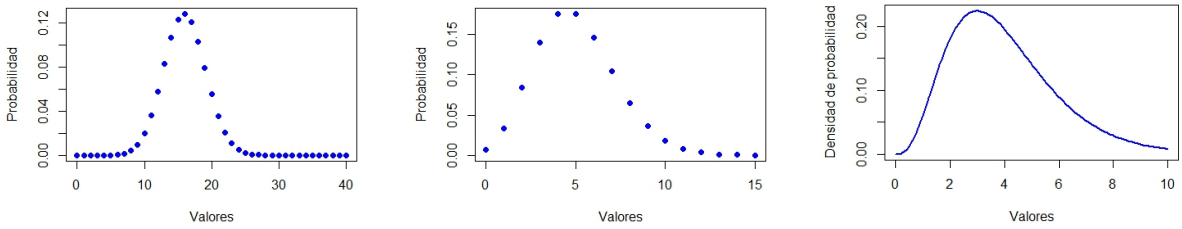
$$f_\theta(y) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right) \tag{1.5}$$

donde θ es conocido como parámetro natural de la distribución, ϕ es el parámetro de escala y a, b y c son funciones arbitrarias. Por simplicidad, consideraremos que todas las observaciones siguen la misma distribución, aunque los resultados que veremos a continuación se pueden generalizar al caso en el que cada observación sigue una distribución de la familia exponencial diferente y estas son independientes entre sí.

En la Figura 1.3 aparecen representadas las funciones de probabilidad y densidad de algunas de las principales distribuciones de la familia exponencial. Más adelante trataremos los casos particulares del modelo logístico y de Poisson y definiremos de forma específica los parámetros θ y ϕ y las funciones a, b y c para la distribución Bernoulli y de Poisson.

La formulación del modelo es la siguiente:

$$\eta = g(\mu) = X_i\beta \tag{1.6}$$



(a) Binomial (b) Poisson (c) Gamma

Figura 1.3: Funciones de probabilidad y densidad de algunas distribuciones procedentes a la familia exponencial.

donde $\mu = \mathbb{E}[Y_i \mid X = x]$ y g es una función conocida como función link o función de enlace.

El planteamiento de este modelo es muy similar al del modelo lineal general, ya que una vez se ha aplicado la función g al parámetro que queremos estimar, se trata de un modelo lineal (o linealizable) en los coeficientes β. De hecho, el modelo lineal general no es más que un caso particular donde la respuesta es normal y la función link es la identidad.

Es posible expresar la esperanza y la varianza de la variable respuesta en términos de las funciones a y b (Wood, 2017, pp. 103-105):

$$\mathbb{E}[Y_i] = b'(\theta)$$
$$\text{Var}(Y_i) = b''(\theta)a(\phi).$$

Si ϕ es conocido, la función a puede tomar cualquier forma. Sin embargo, en la práctica, se desconoce su valor. Por este motivo se suele asumir que $a(\phi) = \frac{\phi}{\omega}$ donde ω es una constante que vale uno en la mayoría de los casos. Estas suposiciones abarcan las principales distribuciones de la familia exponencial y permiten reescribir la varianza de la variable respuesta como:

$$\text{Var}(Y_i) = b''(\theta)\phi.$$

1.2.1. Estimación de los parámetros del modelo

La estimación de los parámetros del modelo se lleva a cabo mediante el método de máxima verosimilitud. La función de verosimilitud es:

$$f_\theta(y_1, ..., y_n) = \prod_{i=1}^{n} f_\theta(y_i) = \prod_{i=1}^{n} \exp\left(\frac{y_i\theta - b(\theta)}{\phi} + c(y_i, \phi)\right).$$

El estimador de máxima verosimilitud de β será aquel que maximice la función de verosimilitud. Para calcularlo, por simplicidad, se trabaja con el logaritmo de la función anterior:

$$l(\beta) = log\left(\prod_{i=1}^{n} f_\theta(y_i)\right) = \sum_{i=1}^{n}\left(\frac{y_i\theta - b(\theta)}{\phi} + c(y_i,\phi)\right). \tag{1.7}$$

La dependencia de θ del vector de parámetros β se explica de forma detallada en la siguiente sección. Por ser un máximo de esta función, el estimador de máxima verosimilitud cumplirá:

$$[\nabla l(\beta)]_j = \frac{\partial l(\beta)}{\partial \beta_j} = 0,\ j = 1, ..., p \tag{1.8}$$

donde $[\nabla l(\beta)]_j$ denota la componente j-ésima del vector gradiente de $l(\beta)$. A estas ecuaciones se las conoce como ecuaciones de verosimilitud y, en general, no tienen solución explícita. Por ello, se recurre a métodos iterativos para su resolución. A continuación, detallaremos la implementación de los procedimientos habituales para la resolución del problema anterior en este caso: el método de Newton [1] y el algoritmo de mínimos cuadrados iterativos.

Algoritmo de Newton

El método de Newton es un procedimiento iterativo que permite resolver las ecuaciones de verosimilitud (1.8). El algoritmo es el siguiente:

$$\beta^{[k+1]} = \beta^{[k]} - (\nabla^2 l(\beta^{[k]}))^{-1}\nabla l(\beta^{[k]}) \tag{1.9}$$

donde $\nabla^2 l(\beta)$ denota la matriz hessiana del logaritmo de la función de verosimilitud.

En primer lugar, calculamos el vector gradiente de $l(\beta)$:

$$[\nabla l(\beta)]_j = \frac{\partial l(\beta)}{\partial \beta_j} = \sum_{i=1}^{n}\frac{1}{\phi}\left(y_i\frac{\partial\theta}{\partial\beta_j} - \frac{\partial b(\theta)}{\partial\beta_j}\right) = \sum_{i=1}^{n}\frac{1}{\phi}\left(y_i\frac{\partial\theta}{\partial\beta_j} - b'(\theta)\frac{\partial\theta}{\partial\beta_j}\right). \tag{1.10}$$

Para calcular esta derivada parcial usamos la regla de la cadena:

$$\frac{\partial\theta}{\partial\beta_j} = \frac{\partial\theta}{\partial\mu}\frac{\partial\mu}{\partial\eta}\frac{\eta}{\partial\beta_j}$$

Obsérvese que η depende de forma lineal de los parámetros del modelo β, como se puede ver en (1.6). Teniendo en cuenta que $\mu = b'(\theta)$ y $\eta = g(\mu)$:

$$\frac{\partial\theta}{\partial\mu} = \frac{1}{\frac{\partial\mu}{\partial\theta}} = \frac{1}{b''(\theta)}$$

$$\frac{\partial\mu}{\partial\eta} = \frac{1}{\frac{\partial\eta}{\partial\mu}} = \frac{1}{g'(\mu)}$$

$$\frac{\partial\eta}{\partial\beta_j} = \frac{\partial}{\partial\beta_j}(\beta_0 + \beta_1 X_{i1} + \ldots + \beta_p X_{ip}) = X_{ij}.$$

[1] Este método ha sido estudiado durante el grado en la materia de Cálculo Numérico en una Variable.

Por tanto,

$$\frac{\partial\theta}{\partial\beta_j} = \frac{X_{ij}}{b''(\theta)g'(\mu)}. \tag{1.11}$$

Sustituyendo (1.11) en (1.10) obtenemos:

$$\begin{aligned}[\nabla l(\beta)]_j &= \frac{1}{\phi}\sum_{i=1}^{n}\left(y_i\frac{X_{ij}}{b''(\theta)g'(\mu)} - b'(\theta)\frac{X_{ij}}{b''(\theta)g'(\mu)}\right) = \\ &= \frac{1}{\phi}\frac{1}{b''(\theta)g'(\mu)}\sum_{i=1}^{n}X'_{ji}(y_i - b'(\theta)) = \\ &= \frac{1}{\phi}\frac{1}{b''(\theta)g'(\mu)}\left[X'(y-\mu)\right]_j. \end{aligned} \tag{1.12}$$

A continuación, calculamos la matriz hessiana de $l(\beta)$. Denotamos por $\left[\nabla^2 l(\beta)\right]_{jk}$ a la componente (j,k) de esta matriz, que se calcula como:

$$\begin{aligned}\left[\nabla^2 l(\beta)\right]_{jk} &= \frac{\partial^2 l(\beta)}{\partial\beta_j\partial\beta_k} = \frac{\partial}{\partial\beta_k}\left(\frac{1}{\phi}\sum_{i=1}^{n}\frac{X_{ij}}{g'(\mu)}\frac{y_i - b'(\theta)}{b''(\theta)}\right) = \\ &= \frac{1}{\phi}\sum_{i=1}^{n}\left\{\frac{\partial}{\partial\beta_k}\left(\frac{X_{ij}}{g'(\mu)}\right)\frac{y_i - b'(\theta)}{b''(\theta)} + \frac{X_{ij}}{g'(\mu)}\frac{\partial}{\partial\beta_k}\left(\frac{y_i - b'(\theta)}{b''(\theta)}\right)\right\}. \end{aligned} \tag{1.13}$$

Definimos la función $V(\mu) = b''(\theta)$ y calculamos estas derivadas parciales mediante la regla de la cadena:

$$\begin{aligned}\frac{\partial}{\partial\beta_k}\left(\frac{X_{ij}}{g'(\mu)}\right) &= \frac{\partial}{\partial\mu}\left(\frac{X_{ij}}{g'(\mu)}\right)\frac{\partial\mu}{\partial\eta}\frac{\partial\eta}{\partial\beta_k} = \frac{-X_{ij}g''(\mu)X_{ik}}{(g'(\mu))^3} \\ \frac{\partial}{\partial\beta_k}\left(\frac{y_i - b'(\theta)}{b''(\theta)}\right) &= \frac{\partial}{\partial\mu}\left(\frac{y_i - b'(\theta)}{V(\mu)}\right)\frac{\partial\mu}{\partial\eta}\frac{\partial\eta}{\partial\beta_k} = -\frac{V(\mu) + V'(\mu)(y_i-\mu)}{(V(\mu))^2}\frac{X_{ik}}{g'(\mu)}. \end{aligned} \tag{1.14}$$

Sustituyendo (1.14) en (1.13):

$$\begin{aligned}\left[\nabla^2 l(\beta)\right]_{jk} &= \frac{1}{\phi}\sum_{i=1}^{n}\left\{\left(\frac{-X_{ij}g''(\mu)X_{ik}}{(g'(\mu))^3}\right)\left(\frac{y_i - b'(\theta)}{b''(\theta)}\right) + \frac{X_{ij}}{g'(\mu)}\left(-\frac{V(\mu) + V'(\mu)(y_i-\mu)X_{ik}}{(V(\mu)^2 g'(\mu)}\right)\right\} = \\ &= -\frac{1}{\phi}\sum_{i=1}^{n}\left\{\frac{X_{ik}X_{ij}}{(g'(\mu))^2 V(\mu)}\left[1 + (y_i-\mu)\left(\frac{g''(\mu)}{g'(\mu)} + \frac{V'(\mu)}{V(\mu)}\right)\right]\right\}. \end{aligned}$$

Definiendo $\alpha(\mu) = 1 + (y_i - \mu)\left(\frac{g''(\mu)}{g'(\mu)} + \frac{V'(\mu)}{V(\mu)}\right)$ y $w_0 = \frac{\alpha(\mu)}{(g'(\mu))^2 V(\mu)}$ podemos reescribir la expresión anterior como:

$$\left[\nabla^2 l(\beta)\right]_{jk} = -\frac{1}{\phi}\sum_{i=1}^{n}\left(\frac{\alpha(\mu)}{(g'(\mu))^2 V(\mu)}X'_{ji}X_{ik}\right) = -\frac{w_0}{\phi}\left[X'X\right]_{jk}. \tag{1.15}$$

Definiendo $g_0 = \frac{g'(\mu)}{\alpha(\mu)}$, podemos expresar (1.12) como:

$$[\nabla l(\beta)]_j = \frac{1}{\phi}w_0 g_0\left[X'(y-\mu)\right]_j. \tag{1.16}$$

Sustituyendo la expresión del vector gradiente (1.16) y de la matriz hessiana (1.15) en (1.9), el algoritmo de Newton se puede expresar como:

$$\begin{aligned}\beta^{[k+1]} &= \beta^{[k]} - \left(-\frac{w_0}{\phi}X'X\right)^{-1}\frac{1}{\phi}w_0 g_0 X'(y-\mu) = \\ &= \beta^{[k]} + (X'X)^{-1}X' g_0(y-\mu) = \\ &= (X'X)^{-1}X'\left[g_0(y-\mu)+\eta\right]\end{aligned}$$

donde la última igualdad es consecuencia de que $\eta = X\beta$. Definiendo una nueva variable $z_i = g_0(y_i - \mu) + \eta$, podemos reescribir el algoritmo como:

$$\beta^{[k+1]} = (X'X)^{-1}X'z. \tag{1.17}$$

Una variante del algoritmo de Newton es el método Fisher Scoring. En él, se utiliza la fórmula (1.9) para calcular los elementos cada iteración, pero se sustituye la matriz hessiana de $l(\beta)$ por su esperanza:

$$\beta^{[k+1]} = \beta^{[k]} - (\mathbb{E}[\nabla^2 l(\beta^{(k)})])^{-1}\nabla l(\beta^{[k]}). \tag{1.18}$$

La esperanza de la componente (j,k) de la matriz hessiana es:

$$\begin{aligned}\mathbb{E}\left[\left[\nabla^2 l(\beta)\right]_{jk}\right] &= -\frac{1}{\phi}\sum_{i=1}^{n}\frac{X_{ij}X_{ik}}{(g'(\mu))^2 V(\mu)}\mathbb{E}\left[1+(Y_i-\mu)\left(\frac{g''(\mu)}{g'(\mu)}+\frac{V'(\mu)}{V(\mu)}\right)\right] = \\ &= -\frac{1}{\phi}\sum_{i=1}^{n}\frac{X_{ij}X_{ik}}{(g'(\mu))^2 V(\mu)}\mathbb{E}\left[\alpha(\mu)\right] = -\frac{1}{\phi}\sum_{i=1}^{n}\frac{X_{ij}X_{ik}}{(g'(\mu))^2 V(\mu)}\end{aligned}$$

donde la última igualdad es consecuencia de que $\mathbb{E}[Y_i - \mu] = \mathbb{E}[Y_i] - \mu = \mu - \mu = 0$. Por tanto, la diferencia entre el algoritmo de Newton clásico y el método de Fisher Scoring es que en este último se toma $\alpha(\mu) = 1$. La principal ventaja de este método con respecto al algoritmo de Newton clásico es que resulta computacionalmente menos costoso. En el caso de observaciones normales, ambos métodos coinciden.

Algoritmo de mínimos cuadrados iterativos

Otra manera de resolver las ecuaciones de verosimilitud (1.8) es mediante el algoritmo de mínimos cuadrados iterativos. En él, en cada una de las iteraciones, se resuelve un problema de mínimos cuadrados en el que la variable respuesta z_i se actualiza tras cada iteración. Esta nueva variable respuesta representa una transformación de la variable respuesta original que sirve para poder compararla directamente con el predictor lineal η. Como se puede observar, este planteamiento se asemeja bastante al ajuste por mínimos cuadrados usual del modelo lineal general. Se procede de la siguiente manera:

1. Inicializar $\hat{\mu} = y_i + \delta_i$ y $\hat{\eta} = g(\hat{\mu})$, donde δ_i es una constante que normalmente se toma como cero o un valor pequeño para garantizar que $g(\hat{\mu})$ es finito.

2. Calcular $z_i = \frac{g'(\hat{\mu})}{\alpha(\hat{\mu})}(y_i - \hat{\mu}) + \hat{\eta}$.

3. Encontrar $\hat{\beta}$ que resuelva el siguiente problema de mínimos cuadrados:

$$\min_{\beta} \sum_{i=1}^{n} (z_i - X_i\beta)^2.$$

4. Verificar si $\hat{\beta}$ cumple el criterio de convergencia $\|\nabla l(\hat{\beta})\| < \varepsilon$ [2], donde ε es una constante pequeña previamente fijada. De ser así, tomamos $\hat{\beta}$ como estimador de máxima verosimilitud de β. Si no, actualizar $\hat{\eta} = X\hat{\beta}$ y $\hat{\mu} = g^{-1}(\hat{\eta})$ y repetir los pasos 2, 3 y 4 hasta la convergencia.

El algoritmo de mínimos cuadrados iterativos es equivalente al método de Newton explicado anteriormente. Esto se puede comprobar fácilmente, ya que la solución al problema de minimización del tercer paso es:

$$\hat{\beta} = (X'X)^{-1}X'z \tag{1.19}$$

que coincide con la fórmula (1.17) para el cálculo de los iterantes del método de Newton.

Estos algoritmos se pueden adaptar al caso general en el que las variables respuesta sean independientes pero no sigan la misma distribución exponencial. En este caso, cada Y_i, $i = 1, ..., n$, seguirá una distribución con función de densidad o probabilidad:

$$f_{\theta_i}(y) = \exp\left(\frac{y\theta_i - b_i(\theta_i)}{a_i(\phi)} + c_i(y, \phi)\right)$$

donde el subíndice i denota los parámetros de la distribución de cada una de las i observaciones. En este caso, las iteraciones del algoritmo de Newton se calculan de la siguiente forma:

$$\beta^{[k+1]} = (X'WX)^{-1}X'Wz \tag{1.20}$$

donde W es una matriz diagonal de orden n que contiene en su diagonal los elementos $w_i = \frac{\alpha(\mu)}{(g'(\mu))^2 V(\mu)}$, $i = 1, ...n$. Obsérvese que este vector se corresponde con la constante w_0 del caso de variables respuesta idénticamente distribuidas.

El algoritmo de mínimos cuadrados iterativo para observaciones independientes e idénticamente distribuidas, recibe en este caso el nombre de algoritmo de mínimos cuadrados ponderados iterativo (IRLS, del inglés iterative re-weighted least squares). La formulación es prácticamente análoga al caso anterior, solo que en el segundo paso hay que actualizar, además de la variable z_i, los pesos w_i. Por otra parte, el problema de minimización que se resuelve en cada una de las iteraciones será un problema de mínimos cuadrados ponderados:

$$\min_{\beta} \sum_{i=1}^{n} w_i(z_i - X_i\beta)^2.$$

[2] $\|\cdot\|$ denota la norma $\mathcal{L}^2$.

La solución de este problema de minimización viene dada por $\hat{\beta} = (X'WX)^{-1}X'Wz$, que coincide con el la fórmula para el cálculo de las iteraciones del algoritmo de Newton que aparece en (1.20).

1.2.2. Medidas de bondad de ajuste

Como ya hemos comentado anteriormente, en el estudio de los modelos de regresión es importante contar con medidas que sirvan para conocer cómo de bien se ajusta el modelo a los datos. Estas medidas se conocen como medidas de bondad de ajuste. Como menciona Green y Silverman (1993), en los modelos lineales generalizados una de las más utilizadas es la *deviance*. Tal y como indica su nombre, mide cuánto se desvían los valores ajustados por el modelo con respecto a las observaciones.

Dado un modelo lineal generalizado con observaciones independientes e idénticamente distribuidas, que llamaremos modelo de referencia, se define su *deviance* escalada como:

$$D^* = 2[l_{\text{máx}} - l_{\text{ref}}]$$

donde $l_{\text{máx}}$ representa el máximo del logaritmo de la función de verosimilitud del modelo saturado y l_{ref} representa el máximo del logaritmo de la función de verosimilitud del modelo de referencia. El modelo saturado es el modelo lineal generalizado que cuenta con tantos parámetros como número de observaciones y, por tanto, es aquel que mejor ajusta los datos. El modelo saturado es de poca utilidad a la hora de ofrecer predicciones y se trata de un modelo muy complejo. Sin embargo, es práctico a la hora de compararlo con otros modelos para determinar la diferencia entre el ajuste de ambos.

Un valor bajo de *deviance* escalada significa que hay poca diferencia entre el ajuste de los datos del modelo saturado y el modelo de referencia y, por tanto, es un indicador de que el modelo de referencia ajusta bien los datos. Por otro lado, un valor alto significa que existe una gran diferencia entre ambos ajustes y, en consecuencia, que el modelo de referencia no se aproxima a los datos. La *deviance* no escalada se define como:

$$\begin{aligned} D = \phi D^* &= 2\phi \left(\sum_{i=1}^{n} \left(\frac{y_i\tilde{\theta} - b(\tilde{\theta})}{\phi} + c(y_i, \phi) \right) - \sum_{i=1}^{n} \left(\frac{y_i\hat{\theta} - b(\hat{\theta})}{\phi} + c(y_i, \phi) \right) \right) = \\ &= 2\sum_{i=1}^{n} \Big(y_i(\tilde{\theta} - \hat{\theta}) - b(\tilde{\theta}) + b(\hat{\theta}) \Big) = 2\sum_{i=1}^{n} d_i \end{aligned} \tag{1.21}$$

donde $\tilde{\theta}$ es parámetro natural evaluado en el estimador de máxima verosimilitud para el modelo saturado, $\hat{\theta}$ es el parámetro natural evaluado en el estimador de máxima verosimilitud para el modelo de referencia y para cada $i = 1, ..., n$:

$$d_i = y_i(\tilde{\theta} - \hat{\theta}) - b(\tilde{\theta}) + b(\hat{\theta}). \tag{1.22}$$

Al estadístico D se lo conoce como *deviance*. La *deviance* es útil porque permite expresar la *deviance* total como suma de las aportaciones individuales de cada una de las observaciones. En el caso de observaciones normales, la *deviance* se reduce a la suma residual de cuadrados.

A partir de la *deviance* podemos definir la *deviance* explicada por el modelo como:

$$D_e = \frac{D_0 - D}{D_0} \tag{1.23}$$

donde D_0 representa el valor de *deviance* del modelo nulo (aquel que está formado únicamente por una constante y no contiene variables explicativas). El modelo nulo será un modelo muy alejado del modelo saturado y, por tanto, su valor de *deviance*, D_0, será muy grande. La *deviance* explicada toma valores entre cero y uno. Un valor próximo a cero significa una *deviance* del modelo considerado (D) muy alta y, por tanto, un mal ajuste de los datos. Un valor próximo a uno indica un valor bajo de *deviance* del modelo considerado y, consecuentemente, un buen ajuste de los datos.

Otra medida de bondad de ajuste del modelo es el estadístico χ^2 de Pearson, que se define como:

$$\chi^2 = \phi \sum_{i=1}^{n} \frac{(y_i - E[Y_i])^2}{\text{Var}[Y_i]} = \sum_{i=1}^{n} \frac{(y_i - \mu)^2}{b''(\theta)} \tag{1.24}$$

donde μ y $b''(\theta)$ se sustituyen por sus respectivas evaluaciones en el estimador de máxima verosimilitud.

Para evaluar de manera local la bondad de ajuste es necesario definir los residuos del modelo. Esto es importante para detectar observaciones atípicas en los datos que podrían ser problemáticas a la hora realizar el ajuste. Los residuos asocian a cada una de las observaciones un valor de discrepancia entre el dato y el valor ajustado por el modelo. Existen varias definiciones de residuo. Los residuos de Pearson se definen como:

$$r_i^{(P)} = \frac{Y_i - \mu}{\sqrt{b''(\theta)}}, \quad i = 1, ..., n.$$

El denominador, $\sqrt{b''(\theta)}$, evaluado en el estimador de máxima verosimilitud, representa una estimación de la desviación típica y sirve para estandarizar los residuos. A partir de la *deviance* se pueden definir los residuos de la *deviance* como:

$$r_i^{(D)} = \text{sign}(Y_i - \mu)\sqrt{di}, \quad i = 1, ..., n \tag{1.25}$$

donde la función sign vale $Y_i - \mu$ si $Y_i - \mu \geq 0$ y $\mu - Y_i$ si $Y_i - \mu < 0$ y d_i es la aportación individual de cada observación a la *deviance* definida en (1.22).

Por otra parte, el estimador de ϕ es:

$$\hat{\phi} = \frac{\chi^2}{n - p}$$

donde $n-p$ representan los grados de libertad del modelo y χ^2 es el estadístico definido en (1.24) evaluado en el estimador de máxima verosimilitud. Este estimador es similar al presentado en (1.3) para el modelo lineal general.

1.2.3. Modelo logístico

Como se puede consultar en Faraway (2006), dos ejemplos importantes de este tipo de modelos son el modelo logístico y el modelo de Poisson. En el caso del modelo logístico, la variable respuesta es binaria, toma los valores cero y uno, y sigue una distribución Bernoulli de parámetro p. Su función de probabilidad es:

$$f(y) = p^y(1-p)^{1-y}, \quad y \in \{0,1\}.$$

Esta función se obtiene tomando, $\mu = p$, $\theta = \log\left(\frac{p}{1-p}\right)$, $a(\phi) = 1$, $b(\theta) = \log(1+\exp(\theta))$ y $c(y,\phi) = 0$ en (1.5).

El objetivo es plantear un modelo capaz de estimar:

$$p = \pi(x) = \mathbb{E}[Y \mid X = x] = P(Y = 1 \mid X = x).$$

Denotamos por $\pi(x)$ a la probabilidad de éxito condicionada a cada valor de la variable X. Al tratarse de una probabilidad, $\pi(x) \in$ [0,1]. Para poder expresarlo como un modelo lineal se debe aplicar antes una función link, que tome valores en la recta real. De esta forma, podemos ajustar el siguiente modelo lineal:

$$g(\pi(x,\beta)) = x'\beta.$$

La función link considerada es la conocida como función logit:

$$g(p) = \log\left(\frac{p}{1-p}\right), \quad \forall p \in [0,1]. \tag{1.26}$$

Al cociente que aparece como argumento del logaritmo se le conoce como odds, y no es más que la probabilidad de éxito entre la probabilidad de fracaso. La odds toma valores en $\mathbb{R}^+$ y al efectuar el logaritmo, conseguimos una cantidad que se sitúa en toda la recta real y que podemos tratar de ajustar con un modelo lineal:

$$\log\left(\frac{1-\pi(x,\beta)}{\pi(x,\beta)}\right) = x'\beta. \tag{1.27}$$

Si queremos expresar la probabilidad de éxito $\pi(x,\beta)$ en función de este modelo lineal, basta considerar la inversa de la función logit,

$$g^{-1}(x) = \frac{e^x}{1+e^x}.$$

Aplicando esta transformación al modelo lineal anterior,

$$\pi(x,\beta) = \frac{e^{x'\beta}}{1+e^{x'\beta}}.$$

1.2.4. Modelo de Poisson

Otro caso particular del modelo lineal generalizado es el modelo de Poisson. En él, la respuesta $Y \in Pois(\lambda)$ y toma valores en $\mathbb{N}$. La función de densidad es:

$$f(y) = \frac{\lambda^y \exp(-\lambda)}{y!}, y = 0, 1, 2, ...$$

Esta función se obtiene a partir de (1.5) tomando $\mu = \lambda$, $\theta = \log(\lambda)$, $a(\phi) = 1$, $b(\theta) = \exp(\theta)$ y $c(y, \phi) = -\log(y!)$.

La función de regresión busca estimar el parámetro:

$$\lambda(x) = \mathbb{E}[Y \mid X = x].$$

Como $\lambda(x)$ es una cantidad estrictamente positiva, es necesario aplicar una función de enlace antes de considerar el modelo lineal. En este caso se toma como función de enlace:

$$g(r) = \log(r), \quad \forall r \in (0, \infty).$$

Aplicando esta función a $\lambda(x)$, conseguimos que tome valores en $\mathbb{R}$ y de esta forma estimarlo con un modelo lineal:

$$g(\lambda(x, \beta)) = x'\beta.$$

Considerando ahora la función inversa de g, $g^{-1}(x)$=e^x, podemos estimar la función de regresión como:

$$\lambda(x, \beta) = e^{x'\beta}.$$

1.2.5. Limitaciones del modelo lineal generalizado

El modelo lineal generalizado, aunque flexibiliza la hipótesis de normalidad del modelo lineal general, sigue siendo insuficiente en algunos contextos. Para ilustrar esto, simulamos unos datos donde la respuesta Y sigue una distribución Bernoulli de probabilidad de éxito $\pi(x)$, de forma que

$$\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = f(x) = 0.3 + 4\sin(2x) + \cos(x) \tag{1.28}$$

donde x toma 300 valores aleatorios en el intervalo [-3,3]. Utilizando la función inversa de logit, la probabilidad de éxito de la variable respuesta Y será:

$$\pi(x) = \frac{e^{f(x)}}{1 + e^{f(x)}} = \frac{e^{0.3+4\sin(2x)+\cos(x)}}{1 + e^{0.3+4\sin(2x)+\cos(2x)}}.$$

Si intentamos ajustar un modelo lineal generalizado (logístico) para estos datos, estaremos intentando estimar la función f, que es claramente no lineal en x, con una función lineal, como indica (1.27). En la Figura 1.4 podemos ver, a la izquierda, la gráfica de la función f frente a la variable explicativa; en el medio, la gráfica de la probabilidad $\pi(x)$ frente a x y, a la derecha, la diagrama de dispersión de la variable respuesta Y frente a la variable explicativa x.

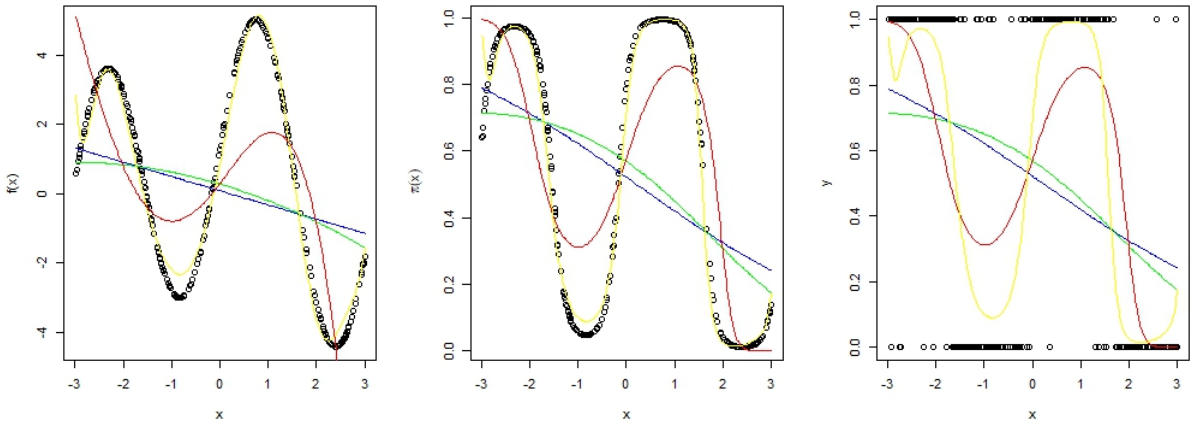


Figura 1.4: Gráfica de f (izquierda) y π (centro) e Y (derecha) frente a la variable explicativa x. Sobre la gráfica de f, los modelos ajustados para f. En azul, el modelo lineal; en verde, el modelo polinómico de grado dos; en rojo, el modelo polinómico de grado cuatro y en amarillo, el modelo polinómico de grado diez. Sobre las gráficas de π e Y, los modelos correspondientes modelos logísticos para π.

Sobre el primer gráfico se encuentra representado el modelo de regresión lineal para f (en azul), el modelo polinómico generalizado de grado dos (en verde), de grado cuatro (en rojo) y de grado diez (en amarillo). Sobre los otros dos gráficos, las correspondientes curvas de regresión logísticas. Podemos observar como cada vez que aumenta el grado del polinomio, los modelos se adaptan mejor a la forma de la función f y consecuentemente a la probabilidad π. Sin embargo, el modelo lineal generalizado, a pesar de ser una generalización del modelo lineal general, sigue siendo ineficiente en este caso. Al igual que en la simulación anterior, el desconocimiento de la forma de la función f obliga a introducir muchos parámetros en el modelo y motiva la búsqueda de métodos más eficientes y generales para su estimación.

La ventaja sobre el modelo lineal general es que el modelo lineal generalizado sí que respeta la naturaleza binaria de la variable Y. La función de regresión $\pi(x)$ no busca estimar directamente los valores de la variable respuesta, sino su probabilidad de éxito condicionada a los valores de la variable explicativa. Como se puede ver en la gráfica central de la Figura 1.4, en todos los modelos ajustados $\pi(x)$ toma valores en el intervalo [0,1]. Por tanto, las predicciones que se

hagan utilizando este modelo consistirán únicamente en ceros o unos, en función de cuál sea la estimación de la probabilidad de éxito para un valor dado de la variable explicativa.

1.3. Introducción a los modelos aditivos

Como comenta Wood (2017), los modelos aditivos aparecen al considerar variables respuesta que no son lineales con respecto a los parámetros del modelo. Sin embargo, se sigue manteniendo la estructura aditiva, es decir, las aportaciones de cada una de las variables se suman entre sí. Su formulación es la siguiente:

$$Y_i = A_i\gamma + f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}) + ... + f_p(x_{pi}), \quad i \in \{1, \dots, n\}$$

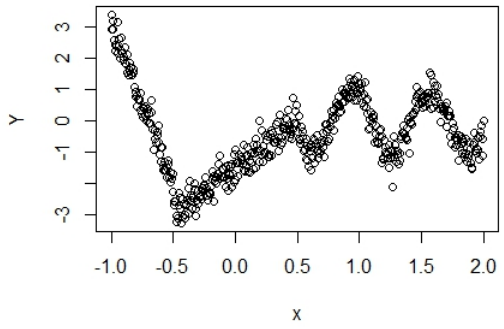
donde A_i es la i-ésima fila de la matriz de diseño correspondiente a las variables explicativas de la parte paramétrica del modelo, γ el vector de parámetros asociado a estas variables y f_l funciones *suaves* de las variables explicativas. La parte paramétrica del modelo permite incluir las variables explicativas que se relacionan de forma lineal con la variable respuesta, tal y como hemos explicado en el modelo lineal general, mientras que las funciones *suaves* modelan las relaciones no lineales entre la respuesta y las covariables.

Estos modelos presentan un problema de identificación, ya que las funciones f_1, f_2,...,f_p y $f_1 + c$, $f_2 - c$,...,f_p (donde c es una constante cualquiera) dan lugar a la misma predicción. Para solventar este problema se imponen la restricciones:

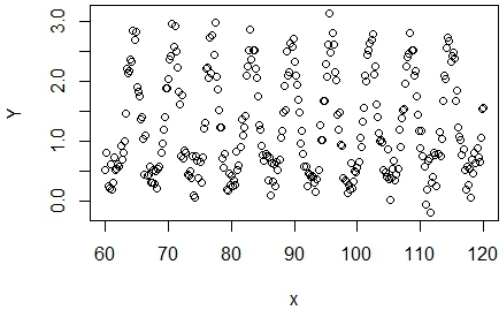
$$\sum_{i=1}^{n} f_l(x_i) = 0 \quad \forall l \in \{1, \dots, p\}. \tag{1.29}$$

Obsérvese que el modelo lineal no es más que un caso particular del modelo aditivo donde las funciones f_l consideradas son lineales. Al igual que en el modelo lineal, las contribuciones de cada una de las variables se suman, pero estas no tienen por qué ser proporcionales a las variables. Esto hace que el modelo sea más flexible pero igualmente fácil de interpretar. La relación con cada una de las variables se modela de forma individual y se puede hacer utilizando distintos enfoques: un ajuste paramétrico, que puede ser lineal o no lineal, o un ajuste no parámetrico. El modelo aditivo generalizado, que se tratará en el siguiente capítulo, combina las hipótesis del modelo lineal generalizado y de los modelos aditivos.

La Figura 1.5 contiene dos ejemplos de datos en los que se puede apreciar a simple vista que la relación entre la variable explicativa y la variable respuesta no es lineal. Además, resultaría muy complejo determinar un modelo linealizable que ajustara bien los datos sin aumentar innecesariamente la complejidad del modelo. Los datos de la Figura 1.5a, se corresponden como un



(a)



(b)

Figura 1.5: Diagramas de dispersión en los que la variable respuesta y la variable explicativa tienen una relación no lineal.

modelo de la forma:

$$Y = \begin{cases} -10x - 7 + \epsilon, & \text{si } x \leq -0.5 \\ 3x - 1.5 + \epsilon, & \text{si } -0.5 < x \leq 0.5 \\ -\cos(10x) + \epsilon, & \text{si } 0.5 < x < 2 \end{cases}$$

donde $\varepsilon \in N(0, 0.3)$ y la variable explicativa toma valores equiespaciados una distancia de 0.01 en el intervalo $[-1.2, 1.2]$. Por otro lado, los datos de la Figura 1.5b proceden de un modelo de la forma: $Y = \log(\sin(x) + 2) + \varepsilon$, $\varepsilon \in N(0, 0.2)$, donde x toma valores equiespaciados una distancia de 0.2 en el intervalo $[60, 120]$. En estos contextos, tienen un papel fundamental los modelos aditivos.

Capítulo 2

Modelos aditivos generalizados: formulación e ideas básicas

En este capítulo partiremos de un modelo con respuesta normal y una única variable explicativa de la siguiente forma:

$$Y_i = f(x_i) + \varepsilon_i, \quad i \in \{1, \ldots, n\} \tag{2.1}$$

donde f es una función *suave* y $\varepsilon_i \in N(0, \sigma^2)$ independientes.

En primer lugar veremos herramientas para la estimación de la función f. Para ello, explicaremos diferentes técnicas como los splines de suavizado, los splines de regresión y los splines de regresión penalizada. Dentro de este último tipo nos centraremos en los P-splines. Ilustraremos, haciendo uso del modelo (1.4), la importancia de la penalización en la estimación por P-splines y explicaremos varios criterios para la elección del parámetro de suavizado. Por último, generalizaremos el modelo (2.1) considerando respuestas que no sean únicamente normales.

2.1. El concepto de spline

A la hora de estimar f en el modelo (2.1), una herramienta útil es expresar esta función en una base de funciones (Perperoglou et al., 2019). Esta base está formada por funciones básicas $B_j(x), \quad j = 1, ..., L$ que se supondrán completamente conocidas.

Por ejemplo, bajo la suposición de que f es una función lineal, podemos usar la base $\{1, x\}$ y expresar f como combinación lineal de estos elementos:

$$Y_i = f(x_i) + \varepsilon_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad i \in \{1, \ldots, n\}.$$

Se puede observar que el modelo anterior no es más que un modelo lineal con intercepto β_0 y pendiente β_1. Sin embargo, como hemos visto, la hipótesis linealidad sobre f es muy restrictiva.

El modelo lineal general admite la introducción de variables que se obtienen mediante operaciones de la variable explicativa original. Por lo tanto, se pueden ajustar multitud de modelos que no sean lineales. Por ejemplo, en un modelo polinómico de orden p, se utiliza la base $\{1, x, x^2, ..., x^p\}$, que permite expresar el modelo de la siguiente forma:

$$Y_i = f(x_i) + \varepsilon_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + ... + \beta_p x_i^p + \varepsilon_i, \quad i \in \{1, \ldots, n\}.$$

No obstante, como ilustramos en el capítulo anterior, el desconocimiento de la forma de la función f hace que sea complicado la elección de una base que aporte la flexibilidad necesaria para la estimación de la función sin aumentar innecesariamente la complejidad del modelo. Además, para poder reflejar los posibles cambios de pendiente sin recurrir a funciones demasiado complicadas, es conveniente considerar f como una función definida a trozos.

Como solución a este problema surgen los splines. Los splines son curvas muy flexibles que resultan de utilidad en la estimación de funciones de las que desconocemos su forma. Consideramos una partición del intervalo $[a, b]$ donde se quiere estimar la función f, dada por J nodos tales que $a < x_1^* < ... < x_J^* < b$. Un spline de grado p es una función polinómica de grado p en cada uno de los $J+1$ subintervalos de $[a, b]$ determinados por la partición, con derivadas continuas en los nodos hasta orden $p-1$.

Por ejemplo, si f es una función spline cúbica, se puede representar mediante un polinomio de grado tres a trozos, con derivadas primeras y segundas continuas en los nodos, es decir:

$$f(x) = a_i(x - x_i^*)^3 + b_i(x - x_i^*)^2 + c_i(x - x_i^*) + d_i \quad \text{para } x \in [x_i^*, x_{i+1}^*].$$

Cada polinomio definido sobre cada uno de los intervalos consta de $p+1$ coeficientes[1]. A esta cantidad se le conoce como orden del spline. Una función polinómica a trozos de grado p definida sobre un intervalo $[a, b]$ con J nodos que determinan $J+1$ intervalos tiene $(p+1)(J+1)$ grados de libertad. Imponiendo las restricciones de continuidad en sus derivadas hasta orden $p-1$ en los nodos, obtenemos $p \cdot J$ restricciones. Por tanto, un spline de grado p definido sobre un intervalo $[a, b]$ con J nodos es un espacio vectorial con $(p+1)(J+1)$- $p \cdot J = J + p + 1$ grados de libertad. Por ejemplo, en un spline de grado tres habrá $J + 4$ parámetros que determinar.

En la Figura 2.1 aparecen representadas tres funciones spline definidas sobre el intervalo [0,5] con nodos en x_1^*=1, x_2^*=1.5, x_3^*=3, x_4^*=4 y x_5^*=4.5. Estos nodos definen seis subintervalos en [0,5]. La Figura 2.1a corresponde a un spline de grado uno. Se trata de una función continua y lineal a trozos. La Figura 2.1b es un spline de grado dos, es decir, una función a trozos formada

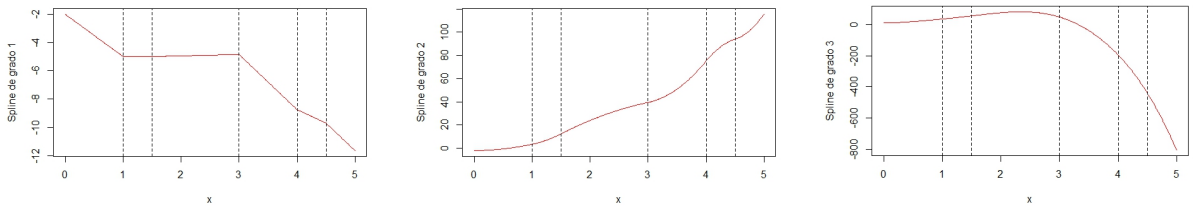
[1]Obsérvese que ahora d_i denota el término independiente de los polinomios y no hace referencia a las aportaciones individuales a la *deviance* definidas en (1.22).

por polinomios de grado dos en cada uno de los subintervalos y con derivadas continuas en los nodos. Por último, la Figura 2.1c representa un spline de grado tres: una función definida a trozos por polinomios de grado tres que cuenta con derivadas continuas hasta orden dos en los nodos.

A mayor número de nodos, la función spline será más flexible y puede adaptarse a formas más extrañas de las nubes de puntos. Sin embargo, una gran cantidad de nodos puede dar como resultado un sobreajuste y una curva que refleje patrones de los datos que son a causa de la aleatoriedad de los mismos y no de una tendencia global. Esta situación aparece ilustrada en la Figura 2.6a que comentaremos posteriormente. Por otra parte, la localización de los nodos también tiene gran importancia en la curva ajustada. El principal problema reside en que no existen criterios sencillos para la elección óptima del número de nodos y de su ubicación (Wood, 2017).

En cuanto al grado del spline, los más utilizados son los de grado tres, ya que es el menor grado para el que se obtienen curvas en las que resulta imperceptible a simple vista la definición a trozos de la función.

En las secciones posteriores veremos dos enfoques distintos que utilizan splines para la estimación de las funciones: los splines de suavizado y los splines de regresión. Por último, explicaremos los splines de regresión penalizada, que surgen como combinación de ambos enfoques. Dentro de ellos, profundizaremos en una de las técnicas más utilizadas hoy en día: los P-splines.



(a) Spline de grado 1 (b) Spline de grado 2 (c) Spline de grado 3

Figura 2.1: Splines de grado uno, dos y tres definidos sobre el intervalo [0,5] con nodos en los valores x_1^*=1, x_2^*=1.5, x_3^*=3, x_4^*=4 y x_5^*=4.5

2.2. Splines de suavización

Los splines de suavización surgen como solución al siguiente problema de minimización:

$$\min_f \sum_{i=1}^{n} (Y_i - f(x_i))^2 + \lambda \int_a^b (f''(x))^2 \, dx.$$

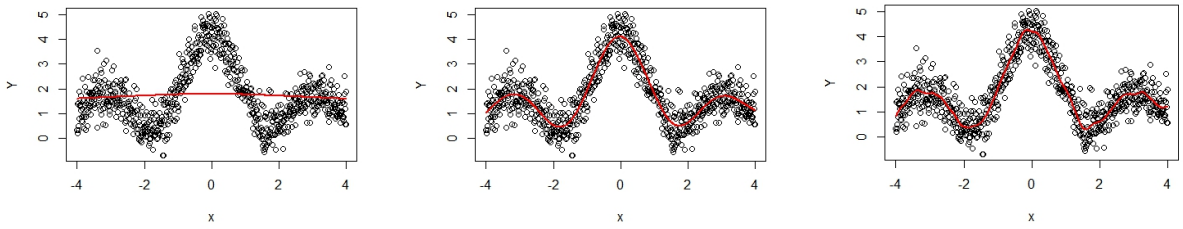
Se trata de encontrar una función f que minimice la suma de residuos al cuadrado a la que se le añade una penalización para la oscilación de la curva expresada en términos de la derivada segunda. El valor λ es conocido como parámetro de suavizado y controla el peso de esta penalización. Para resolver el problema, los valores $f(x_i)$ se tratan como n parámetros a determinar. La solución es un spline cúbico natural con nodos en los valores x_i. Por tanto, el número de nodos será igual al número de observaciones, $J = n$ (Wood, 2017).

Un spline cúbico natural es un spline cúbico con segunda y tercera derivada nulas en los nodos de los extremos, es decir, $f''(x_1^*) = f''(x_J^*) = f'''(x_1^*) = f'''(x_J^*) = 0$. Se trata de un spline cúbico que es lineal en el primer y último intervalo definido por los nodos. Imponiendo estas cuatro restricciones a un spline cúbico, el número de parámetros a determinar de un spline cúbico natural es $J + 3 + 1 - 4 = J = n$. Como comenta Wood (2017), estos splines solventan el problema de la elección del número de nodos y de su localización pero son computacionalmente muy costosos. El gran inconveniente reside en el elevado número de parámetros a determinar, ya que hay tantos como número de observaciones.

Los splines de suavización controlan el equilibrio entre la suavidad de la curva y el buen ajuste del modelo a través de la penalización de la función objetivo. A mayores valores de λ, la curva ajustada será más suave pero estará más alejada de los datos. Para valores pequeños, será mas oscilante pero ajustará mejor los datos. Estas situaciones se pueden observar en la Figura 2.2, donde se ilustran los diagramas de dispersión del modelo (1.4) junto con los modelos ajustados mediante splines de suavización para tres valores del parámetro de suavizado distintos. En la Figura 2.2a, para la cual se ha tomado λ=10, se puede ver como la curva ajustada no se adapta en absoluto a la nube de puntos. Esto es debido al alto peso de la penalización, que provoca una curva demasiado suave. Por el contrario, en la Figura 2.2c, para la que se ha elegido λ=10^{-5}, se puede ver como el modelo sobreajusta los datos, dando lugar a una curva con demasiadas oscilaciones. En la Figura 2.2b, en la que se ha considerado $\lambda = 0.001$, se ha conseguido una curva suave, sin oscilaciones, que determina bastante bien la forma de la nube de puntos. En la sección 2.5 veremos un criterio para la elección óptima de este parámetro.

2.3. Splines de regresión

La idea fundamental de los splines de regresión es representar la función f a través de una base de splines. Como hemos visto anteriormente, el conjunto de splines de grado p definidos sobre un intervalo $[a, b]$ con J nodos forman un espacio vectorial de dimensión $J + p + 1$. Consideramos una base de splines formada por funciones básicas $B_j(x)$, $j = 1, ..., J + p + 1$ y suponemos que



(a) λ=10 (b) λ=0.001 (c) λ=0.00001

Figura 2.2: Diagrama de dispersión del modelo (1.4) junto con las curvas ajustadas mediante splines de suavizado para distintos valores de λ.

f admite representación en esta base. Entonces, se puede expresar como:

$$f(x) = \sum_{j=1}^{J+p+1} \beta_j B_j(x).$$

Esta representación de f permite reescribir el problema de minimización de los residuos al cuadrado como:

$$\min_f \sum_{i=1}^{n} (Y_i - f(x_i))^2 = \min_\beta \sum_{i=1}^{n} (Y_i - \sum_{j=1}^{J+p+1} \beta_j B_j(x_i))^2.$$

Se trata de un problema de mínimos cuadrados usual en el se consideran $B_j(x)$ como nuevas variables. Los splines de regresión tienen muchas similitudes con la estimación paramétrica de los modelos que hemos visto hasta ahora, ya que la estimación de la función f se reduce a la estimación de los parámetros β_j . La solución de este problema de minimización permite estimar f como:

$$\hat{f}(x) = \sum_{j=1}^{J+p+1} \hat{\beta}_j B_j(x).$$

La flexibilidad de la curva ajustada se controla a través del número de nodos considerados y de sus localizaciones, como veremos después en la Figura 2.5. A mayor número de nodos, la curva ajustará mejor a los datos pero será menos suave. La principal desventaja de este tipo de splines reside en que no se conocen ningún criterio global y sencillo para la elección óptima del número de nodos ni para sus localizaciones (Wood, 2017).

La elección de la base de splines no modifica demasiado el ajuste de la curva. Como explica Perperoglou et al. (2019), las diferencias entre las distintas bases se encuentran en su estabilidad numérica. La elección de una base con mala estabilidad numérica podría provocar que pequeñas variaciones en los datos supusieran grandes variaciones en la estimación de la función. A continuación, explicaremos dos bases distintas de splines: los polinomios truncados y los B-splines.

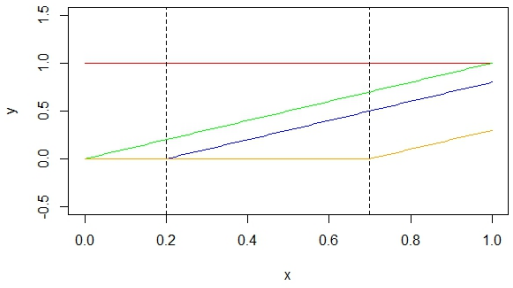
Dados J nodos $x_1^*, ..., x_J^*$ del intervalo $[a, b]$, una base de polinomios trucados de grado p está formada por:

$$\begin{aligned}
&B_1(x) = 1,\\
&B_2(x) = x,\\
&\vdots\\
&B_{p+1}(x) = x^p,\\
&B_j(x) = (x - x_j^*)_+^p \quad \text{para } j = p+2, \ldots, J+p+1\\
&\text{donde} \quad u_+^p = (u^p)_+ = \begin{cases} u^p & \text{si } u \geq 0 \\ 0 & \text{si } u < 0 \end{cases}
\end{aligned}$$

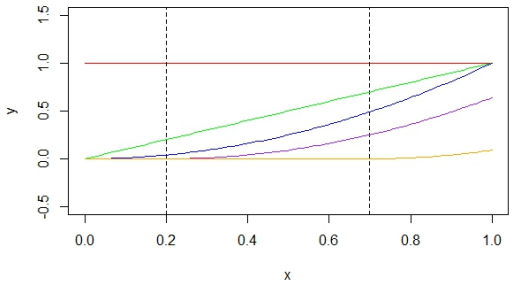
Estas funciones permiten expresar la función f como:

$$f(x) = \beta_0 + \beta_1 x + ... + \beta_p x^p + \sum_{j=1}^{J} \beta_j (x - x_j^*)_+^p$$

En la Figura 2.3 aparecen representadas las funciones de una base de polinomios truncados de grado uno (Figura 2.3a) y grado dos (Figura 2.3b), definidas en el intervalo [0,1] con nodos en en $x_1^* = 0.2$ y $x_2^* = 0.7$. En la Figura 2.3a, las funciones de la base son: $B_1(x) = 1$ (en rojo); $B_2(x) = x$ (en verde); $B_3(x) = (x-0.2)_+$ (en azul) y $B_4(x) = (x-0.7)_+$ (en amarillo). Obsérvese que el tamaño de la base es cuatro, es decir, $J + p + 1$, donde $J = 2$ y $p = 1$. De forma análoga se definen las funciones de la Figura 2.3b.



(a) Funciones de una base de polinomios truncados de grado uno



(b) Funciones de una base de polinomios truncados de grado dos

Figura 2.3: Funciones de una base de polinomios truncados definidas en el intervalo [0,1] con nodos en $x_1^* = 0.2$ y $x_2^* = 0.7$.

A pesar de su fácil interpretación e implementación, las bases de polinomios truncados tienen como principal desventaja la colinealidad entre las funciones de la base. Como se puede ver en la Figura 2.3, estas funciones se solapan entre sí, lo que hace que puedan surgir problemas de

colinealidad entre ellas y de inestabilidad numérica a la hora de estimar el modelo. Las bases de B-splines buscan solucionar este problema mediante funciones que son nulas en todo punto salvo en un intervalo, de forma que se superpongan lo menos posible.

Para definir las funciones de una base de B-splines empezaremos por considerar una partición del intervalo $[a, b]$ dada por J nodos interiores, $x_1^*, \dots x_J^*$. Denotaremos por x_0^* y x_{J+1}^* a los nodos extremos a y b, respectivamente. Existen muchas formas de distribuir los nodos en el intervalo. Lo más habitual es considerar nodos equidistantes o posicionarlos en los cuantiles de la variable explicativa, consiguiendo así que cada intervalo de la partición cuente con el mismo número de observaciones. Añadimos a esta partición p nodos a la izquierda de x_0^*, que tengan el mismo valor que x_0^*, y p nodos a la derecha de x_{J+1}^*, que tengan el mismo valor que x_{J+1}^*. Esto es necesario para definir de forma correcta las funciones básicas B-spline. Por tanto, la nueva partición cuenta con $J + 2p + 2$ nodos. A partir de la función:

$$B_j^0(x) = \begin{cases} 1, & \text{si } x_j^* \leq x < x_{j+1}^* \\ 0, & \text{en otro caso} \end{cases}$$

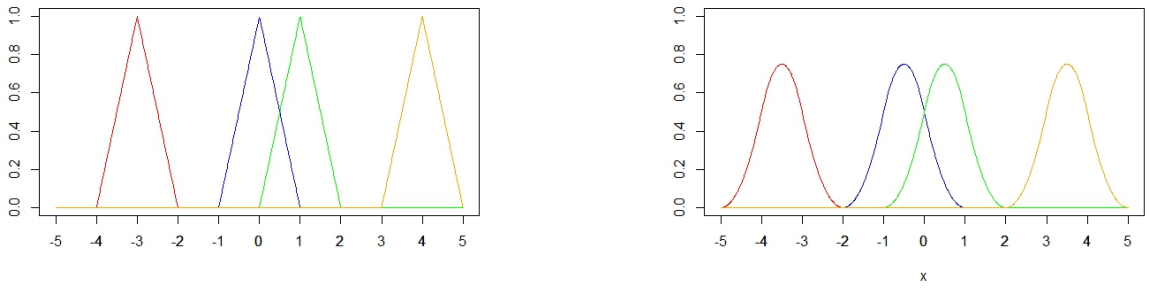
donde $B_j^0(x) = 0$ si $x_j^* = x_{j+1}^*$, podemos definir de forma recursiva las funciones de la base de B-splines de grado p, para $j = 1, \dots, J + p + 1$:

$$B_j^p(x) = \frac{x - x_j^*}{x_{j+p}^* - x_j^*} B_j^{p-1}(x) - \frac{x_{j+p+1}^* - x}{x_{j+p+1}^* - x_{j+1}^*} B_{j+1}^{p-1}(x).$$

En la Figura 2.4 aparecen representadas algunas funciones básicas de dos bases distintas de B-splines. Como se puede observar, estas funciones se superponen menos que las funciones básicas de polinomios truncados. Esta propiedad hace que se reduzca la colinealidad de las funciones de la base y que los métodos de estimación de la función f sean más estables numéricamente cuando utilizamos bases de B-splines. En la Figura 2.4a podemos ver representadas funciones de una base de B-splines de grado uno. Cada una estas funciones está formada por dos rectas: una de ellas definida en el intervalo $[x_j^*, x_{j+1}^*]$ y la otra en $[x_{j+1}^*, x_{j+2}^*]$. En el resto de intervalos definidos por los nodos, vale 0.

En la Figura 2.4b están representadas funciones de una base de B-splines de grado dos. Cada una de las funciones está formada por tres trozos de polinomios cuadráticos: el primero de ellos definido en $[x_j^*, x_{j+1}^*]$; el segundo en $[x_{j+1}^*, x_{j+2}^*]$ y el tercero en $[x_{j+2}^*, x_{j+3}^*]$. Estos polinomios se unen entre sí en los dos nodos interiores del intervalo $[x_j^*, x_{j+3}^*]$. Al considerar una base formada por B-splines, estamos escalando y sumando los B-splines entre sí, dando lugar a multitud de curvas distintas.

En la Figura 2.5 podemos observar la relevancia de la elección de la base y de los nodos en el ajuste del modelo (1.4). En Figura 2.5a aparece representadas las curvas obtenidas utilizando



(a) Funciones básicas B-spline de grado uno.

(b) Funciones básicas B-spline de grado dos.

Figura 2.4: Representación gráfica de algunas de las funciones de las bases de B-splines de grado uno y dos definidas en el intervalo [-5,5] con nodos en los valores enteros de ese intervalo.

splines de regresión con bases de polinomios truncados de grado tres considerando cuatro nodos (en rojo) y doce nodos (en verde). De la misma forma, la Figura 2.5b contiene los ajustes del modelo utilizando splines de regresión con bases de B-splines de grado tres con cuatro nodos (en rojo) y doce nodos (en verde). En ambas gráficas se puede apreciar claramente la diferencia entre las curvas en rojo y en verde. Al aumentar el número de nodos considerado, el modelo se ajusta mejor a los datos. Sin embargo, si comparamos las Figuras 2.5a y 2.5b, podemos observar que los modelos ajustados para el mismo número de nodos son prácticamente iguales a pesar de utilizar bases distintas. En resumen, en la estimación del modelo mediante splines de regresión, la flexibilidad de la curva ajustada depende casi exclusivamente de la elección de los nodos y del grado de los polinomios de la base. Normalmente se utilizan polinomios de grado tres, ya que generan curvas *suaves* y la diferencia con grados superiores es mínima (Perperoglou et al., 2019).

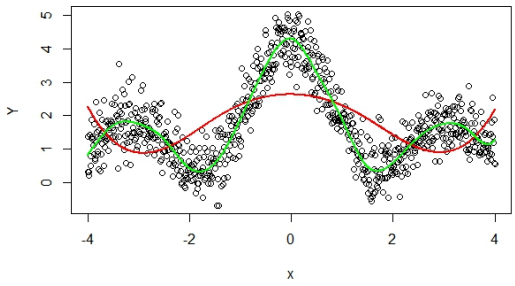
2.4. Regresión spline penalizada

Como explica Perperoglou et al. (2019), los splines de regresión penalizada combinan los dos enfoques anteriores: por una parte, añaden a la función objetivo una penalización para la oscilación de la curva y por otra, utilizan bases de splines para la representación de la función f.

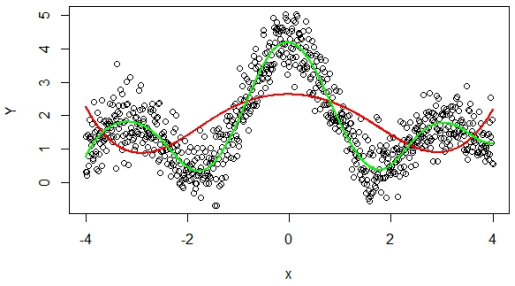
Los splines de regresión penalizada buscan minimizar en β (los coeficientes de f en la base de splines):

$$\min_f \sum_{i=1}^{n}(Y_i - f(x_i))^2 + \lambda \int_a^b (f''(x))^2\,dx \approx \min_\beta \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{J+p+1} \beta_j B_j(x_i))^2 + \lambda \beta^T P \beta$$

donde P es una matriz cuadrada de orden $J + p + 1$ de penalizaciones para la oscilación de la



(a) Ajuste del modelo considerando una base de polinomios truncados de grado tres con cuatro nodos (rojo) y doce nodos (verde)



(b) Ajuste del modelo considerando una base de B-splines de grado tres con cuatro nodos (rojo) y doce nodos (verde)

Figura 2.5: Diagramas de dispersión del modelo (1.4) junto con distintos ajustes obtenidos mediante splines de regresión.

curva y λ es el parámetro de suavizado. La solución a este problema minimiza, por una parte, los residuos al cuadrado y, por otra, la irregularidad de la curva. Dado λ, la solución del problema de minimización es:

$$\hat{\beta} = (X^T X + \lambda P)^{-1} X^T Y$$

donde X es la matriz que contiene en la posición (i,j) el elemento $B_j(x_i)$, $i = 1, ..., n$ y $j = 1, .., J + p + 1$. Obsérvese que si definimos:

$$H = X(X^T X + \lambda P)^{-1} X^T \tag{2.2}$$

podemos reescribir las predicciones del modelo como: $\hat{Y} = X\beta = HY$. La matriz H es una matriz de proyección de orden n que, a diferencia del modelo lineal general sin penalizaciones, no es idempotente.

Un tipo importante de splines de regresión penalizados son los P-splines (Eilers et al., 2015). Los P-splines utilizan bases de B-splines y una penalización discreta que se basa en fórmulas de diferencias de orden d en los coeficientes β_j y que resultan muy fáciles de calcular. Se penaliza los coeficientes cuya diferencia de orden d al cuadrado es grande, es decir, que están muy separados entre sí. De esta forma, se fuerza que las diferencias entre los coeficientes sean pequeñas y, como consecuencia, se consigue un ajuste más suave y con menos oscilaciones. Denotamos por $\Delta^d \beta_j$ la diferencia de orden d de β_j. Esta se calcula de manera recursiva como:

$$\begin{aligned} \Delta\beta_j &= \beta_j - \beta_{j-1} \\ \Delta^2\beta_j &= \Delta(\Delta\beta_j) = \beta_j - 2\beta_{j-1} + \beta_{j-2} \\ &\vdots \\ \Delta^d\beta_j &= \Delta(\Delta^{d-1}\beta_j) \end{aligned} \tag{2.3}$$

Denotamos por D_d una matriz tal que $D_d\beta = \Delta^d\beta$, siendo $\Delta^d\beta$ la matriz columna que contiene en cada una de sus filas la diferencias de orden d del elemento β_j.

Los P-splines consideran como matriz de penalizaciones $P = D_d^T D_d$. El modelo se ajustará minimizando en β:

$$\min_{\beta} \sum_{i=1}^{n} (Y_i - \sum_{j=1}^{J+p+1} \beta_j B_j(x_i))^2 + \lambda \beta^T D_d^T D_d \beta \tag{2.4}$$

Normalmente se toma $d = 2$. La matriz de diferencias de orden 2 es:

$$D_2 = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots \\ 0 & 1 & -2 & 1 & \cdots \\ 0 & 0 & 1 & -2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

De forma que:

$$\beta^T D_2^T D_2 \beta = (\beta_1 - 2\beta_2 + \beta_3)^2 + ... + (\beta_{k-2} - 2\beta_{k-1} + \beta_k)^2$$

En la estimación mediante P-splines se consideran nodos equiespaciados y un tamaño de base suficientemente grande para conseguir una buena aproximación de la curva. La suavidad de la curva ajustada se controla mediante la penalización.

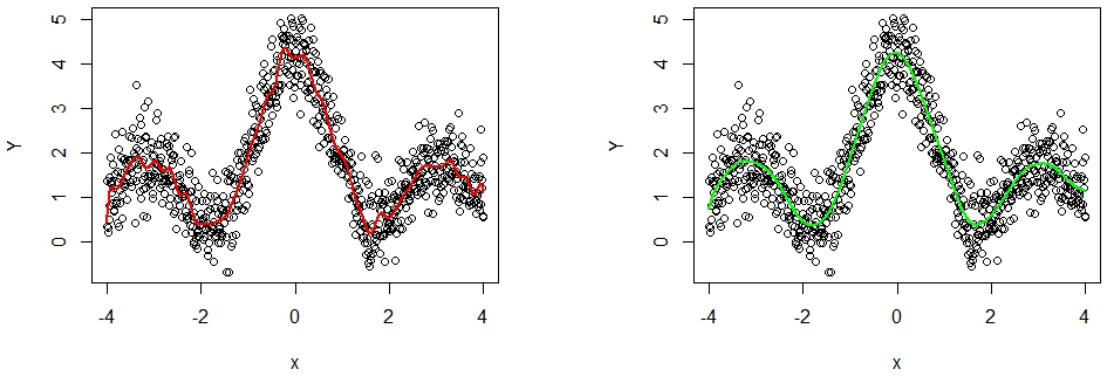
Por tanto, los P-splines solventan el problema de la elección del número de nodos y de su localización. Sin embargo, al igual que en los splines de suavizado, surge la pregunta de cómo elegir el parámetro de suavizado óptimo.

La elección de λ será determinante en la estimación de la función f en el modelo (2.1). Para un número de nodos suficientemente grande, la dimensión de la base tendrá muy poca influencia en la forma de la función, que dependerá casi de forma exclusiva de la penalización de la función objetivo. El parámetro de suavizado controla el equilibrio entre la suavidad de la curva y el ajuste de los datos. Si $\lambda = 0$, estaremos ajustando un modelo sin penalizaciones. Si además utilizamos un gran número de nodos se producirá un sobreajuste de los datos y dará lugar a una curva irregular. Por otra parte, cuando $\lambda \to \infty$, la curva tenderá a un ajuste lineal.

La Figura 2.6 ilustra cómo afecta la elección parámetro de suavizado a la estimación de la curva mediante regresión spline penalizada. En ella, se representa el ajuste del modelo (1.4) utilizando P-splines con dos parámetros de suavizado distintos. Para este ajuste se ha utilizado una base de B-splines cúbicos con 48 nodos equiespaciados en el intervalo [-4,4] y penalizaciones basadas en diferencias de orden 2.

En la Figura 2.6a se ha tomado $\lambda = 0.001$, mientras que en la Figura 2.6b se ha considerado $\lambda = 0.10$. Se puede apreciar como para el valor más pequeño de λ la curva es más irregular,

mientras que para el valor de λ mayor, la gráfica de la estimación de f tiene una forma más suave y menos oscilante. El objetivo es elegir un parámetro de suavizado para el cual el modelo sea capaz de captar el patrón que sigue el diagrama de dispersión pero no las oscilaciones puntuales debidas a la aleatoriedad del error.



(a) λ=0.001 (b) λ=10

Figura 2.6: Diagrama de dispersión del modelo (1.4) junto con las curvas ajustadas mediante P-splines para dos valores distintos del parámetro de suavizado λ.

2.5. Elección del parámetro de suavizado

Uno de los métodos más utilizados para la selección del parámetro de suavizado, tanto para los splines de suavización como los splines de regresión penalizada, es el criterio de validación cruzada ordinaria (Eilers & Marx, 1996). Este criterio busca minimizar en λ:

$$VCO(\lambda) = \sum_{i=1}^{n} (Y_i - \hat{f}_{-i}(x_i; \lambda))^2 \tag{2.5}$$

donde $\hat{f}_{-i}(x_i; \lambda)$ representa la estimación de la función f obtenida mediante mínimos cuadrados penalizados con parámetro de suavizado λ y utilizando el conjunto de datos en el cual se ha suprimido el par (x_i,Y_i). El criterio de validación cruzada mide como predice el modelo un dato faltante utilizando el resto de datos. El valor de λ óptimo será el que minimice la suma de los residuos al cuadrado del modelo ajustado omitiendo el dato (x_i,Y_i), i=1,...,n.

Debido al elevado coste computacional que supone ajustar n modelos distintos, especialmente cuando el número de datos es alto, (2.5) se puede reescribir como:

$$VCO(\lambda) = \sum_{i=1}^{n} \left(\frac{Y_i - \hat{f}(x_i; \lambda)}{1 - h_{ii}} \right)^2 \tag{2.6}$$

donde h_{ii} es el elemento (i, i) de la matriz H definida en (2.2). Este valor se conoce *leverage* del dato i-ésimo y mide la capacidad de influencia de la observación i-ésima sobre el ajuste del modelo. Se puede demostrar fácilmente que las ecuaciones (2.5) y (2.6) son equivalentes.

Una variación del criterio de validación cruzada ordinaria es el criterio de validación cruzada generalizada. Este criterio se ve menos afectado por observaciones demasiado influyentes al sustituir el valor h_{ii} por $\frac{tr(H)}{n}$, el valor promedio de los *leverages* de todas las observaciones.

Denotamos por $tr(H)$ la traza de la matriz H. Esta representará los grados de libertad del modelo (que representaremos por df):

$$df = tr(H) \tag{2.7}$$

Bajo este criterio, el parámetro de suavizado óptimo es aquel que minimiza en λ:

$$VCG(\lambda) = \sum_{i=1}^{n} \left(\frac{Y_i - \hat{f}(x_i; \lambda)}{1 - \frac{tr(H)}{n}} \right)^2 \tag{2.8}$$

Análogamente al caso del modelo lineal generalizado, bajo la suposición de homocedasticidad, se puede estimar la varianza residual como:

$$\hat{\sigma}^2 = \frac{1}{n - tr(H)} \sum_{i=1}^{n} (Y_i - \hat{f}(x_i))^2 = \frac{\text{RSS}}{n - tr(H)}$$

Sin embargo, a diferencia del modelo lineal generalizado, $\hat{\sigma}$ no es un estimador insesgado (Eilers et al., 2015).

2.6. Regresión spline penalizada para respuesta generalizada

Los modelos aditivos generalizados aparecen como combinación de los modelos lineales generalizados y los modelos aditivos. Por una parte, permiten que la respuesta no sea normal, y, a su vez, se relajan las restricciones sobre la relación entre las variables explicativas y la variable respuesta.

Consideraremos, en primer lugar, un modelo sencillo con una única variable explicativa. Asumimos las observaciones Y_i son independientes entre sí e idénticamente distribuidas con

función de densidad o probabilidad de la forma (1.5), es decir, pertenecen a una distribución de la familia exponencial. El modelo se puede formular como:

$$g(\mu) = f(x_i), \quad i \in \{1, ..., n\}$$

donde $\mu = \mathbb{E}[Y_i \mid X = x]$. Utilizando una base de splines formada por funciones $B_j(x)$, $j = 1, ..., J + p + 1$, podemos expresar el modelo anterior como:

$$g(\mu) = \sum_{j=1}^{J+p+1} \beta_j B_j(x_i), \quad i \in \{1, ..., n\}$$

Si denotamos por X a la matriz que contiene en la posición (i,j) el elemento $B_j(x_i)$, el modelo se puede reescribir como:

$$g(\mu) = X_i\beta, \quad i \in \{1, ..., n\}$$

donde X_i representa la fila i-ésima de esta matriz. De esta forma, conseguimos una formulación análoga a la del modelo de regresión lineal generalizado expuesto anteriormente en (1.6).

Como comenta Wood (2017), en la estimación mediante splines de regresión penalizada con respuesta generalizada, la estimación de los coeficientes de la base se lleva a cabo mediante el método de máxima verosimilitud penalizada. El estimador de máxima verosimilitud penalizado será aquel que maximice en β el logaritmo de la función de verosimilitud penalizada:

$$l_p(\beta) = l(\beta) - \frac{1}{2\phi}\lambda\beta' P\beta$$

donde $l(\beta)$ es el logaritmo de la función de máxima verosimilitud no penalizada definido en (1.7).

Dado β, la maximización de esta función se lleva a cabo mediante el método de mínimos cuadrados penalizados iterativos. Para ello, se procede de la siguiente manera:

1. Inicializar $\hat{\mu} = y_i + \delta_i$ y $\hat{\eta} = g(\hat{\mu})$, donde δ_i es una constante que normalmente se toma como cero o un valor pequeño para garantizar que $g(\hat{\mu})$ es finito.
2. Calcular $z_i = \frac{g'(\hat{\mu})}{\alpha(\hat{\mu})}(y_i - \hat{\mu}) + \hat{\eta}$.
3. Encontrar $\hat{\beta}$ que resuelva el siguiente problema de mínimos cuadrados:

$$\min_{\beta} \sum_{i=1}^{n} (z_i - X_i\beta)^2 + \lambda\beta' P\beta$$

4. Verificar si $\hat{\beta}$ cumple el criterio de convergencia $\|\nabla l_p(\hat{\beta})\| < \varepsilon$, donde ε es una constante pequeña previamente fijada. De ser así, tomamos $\hat{\beta}$ como estimador de máxima verosimilitud de β. Si no, actualizar $\hat{\eta} = X\hat{\beta}$ y $\hat{\mu} = g^{-1}(\hat{\eta})$ y repetir los pasos 2, 3 y 4 hasta la convergencia.

Al igual que en el caso del modelo lineal generalizado, se podrían considerar observaciones que no procedieran de la misma familia exponencial. El planteamiento sería análogo al hecho en el capítulo uno. En ese caso, el algoritmo se conoce como mínimos cuadrados ponderados iterativos penalizados (PIRLS, del inglés penalized iteratively re-weighted least squares).

En el caso más general del modelo aditivo generalizado se considera una respuesta generalizada y varias variables explicativas que se relacionan con la variables respuesta a través de la funciones *suaves* f_k, $k = 1, ...l$. Se puede formular como:

$$g(\mu) = A_i\gamma + f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}) + ... + f_l(x_{li}), \quad i \in \{1, \dots, n\} \tag{2.9}$$

Obsérvese que la matriz de diseño de la parte paramétrica del modelo se denota ahora por A, siendo A_i su fila i-ésima. Los coeficientes de la parte paramétrica del modelo se representan por γ.

La representación de las funciones en la base y la introducción de penalizaciones a la oscilación se realiza de forma individual para cada una de las funciones que aparecen en el modelo. Denotaremos con un superíndice k a los parámetros vectores y matrices relativos a la función f_k. Suponemos que cada f_k se puede representar en una base de splines de grado p^k definida en un intervalo $[a, b]$ con J^k nodos mediante unos coeficientes β_j^k:

$$f_k(x) = \sum_{j=1}^{J^k+p^k+1} \beta_j^k B_j^k(x)$$

Sea χ^k la matriz que contiene en el elemento (i, j) los valores $B_j^k(x_i)$, $i = 1, ..., n$, $j = 1, ..., J^k + p^k + 1$. Para cada una de las funciones se considera una penalización de la forma:

$$\lambda^k(\beta^k)' P^k \beta^k$$

Para garantizar que las funciones f_k son las únicas que aportan una predicción determinada Y_i se imponen las restricciones de identificación definidas en (1.29). Estas restricciones se incluyen en el modelo a través de una reparametrización de de la base, como se puede consultar en (Wood, 2017, p. 211). Denotamos por X^k la matriz χ^k que incluye la restricción de identificación.

La matriz del modelo (2.9) estará formada por la columnas de la matriz de la parte paramétrica del modelo A y de cada una de las matrices X^k, $k = 1, ..., l$:

$$X = \left(A \,\middle|\, X^1 \,\middle|\, X^2 \,\middle|\, \cdots \,\middle|\, X^l\right)$$

El vector de parámetros del modelo estará formado por los coeficientes de la parte paramétrica del modelo junto con los coeficientes de cada función f_k en cada una de las bases:

$$\beta = \begin{pmatrix} \gamma & \beta_1^1 & \beta_2^1 & \cdots & \beta_{J^k+p^k+1}^1 & \cdots & \beta_1^l & \beta_2^l & \cdots & \beta_{J^l+p^l+1}^l \end{pmatrix}'$$

Tras estas consideraciones, nuestro modelo se puede reescribir como

$$g(\mu) = X_i\beta \tag{2.10}$$

El ajuste se llevará a cabo de manera análoga al caso unidimensional, mediante la maximización de la función de verosimilitud:

$$l_p(\beta) = l(\beta) - \frac{1}{2\phi}\sum_{k=1}^{l}(\beta^k)'P^k\beta^k$$

Para la resolución de este problema se utilizará el algoritmo de mínimos cuadrados penalizados iterativos explicado anteriormente considerando, en el tercer paso, el problema de mínimos cuadrados:

$$\min_{\beta}\sum_{i=1}^{n}(z_i - X_i\beta)^2 + \sum_{k=1}^{l}(\beta^k)'P^k\beta^k$$

Al igual que en el caso anterior, se podrían considerar observaciones independientes no idénticamente distribuidas siempre que sigan distribuciones de la familia exponencial. Entonces, la maximización se llevaría a cabo mediante el algoritmo de mínimos cuadrados ponderados iterativos penalizados (PIRLS).

2.7. Ejemplo ilustrativo con datos simulados

En esta sección utilizaremos los datos simulados por el modelo (1.28) con el propósito de ilustrar algunos de los resultados vistos a lo largo del trabajo en un ejemplo con respuesta generalizada. Para ello, utilizaremos la función `gam`, que se incluye en el paquete `mgcv` de R. En Wood (2023) se puede encontrar documentación sobre este paquete. Un modelo aditivo generalizado para estos datos se podría formular de la siguiente manera:

$$\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + f(x) \tag{2.11}$$

donde $\pi(x)$ representa la probabilidad de éxito de la variable respuesta condicionada a cada valor de la variable explicativa, β_0 es el intercepto (y en este caso, la única parte paramétrica del modelo) y f es una función *suave* de la variable explicativa. Este modelo es de la forma (2.9), donde la función g es la función logit definida en (1.26). Mediante el siguiente código:

```
mod_gam <- gam(y ~ s(x, k = -1, bs = "ps"), family = binomial(link = "logit"),
               method = "GCV.Cp")
```

estamos ajustando el modelo aditivo generalizado (2.11) utilizando regresión spline penalizada. La variable explicativa se incluye dentro del término de suavizado, el cual se denota por la letra

"s" (que se corresponde con f en la fórmula 2.11). Dentro de este término, el argumento `k=-1`, indica que se tomará el tamaño de base por defecto para estimar la función (el cual depende de la base considerada). Como hemos visto, en el caso de la estimación mediante regresión spline penalizada, se opta por considerar un tamaño de base suficientemente grande para conseguir un buen ajuste y controlar la suavidad de la curva mediante una penalización, por lo que en este caso no tendrá demasiada importancia este valor. Mediante `bs=''ps''` estamos especificando que se utilicen P-splines para la estimación de la función (una base de B-splines cúbicos y penalizaciones discretas basadas en diferencias de orden dos). Para esta base, la dimesión por defecto es 10. Por último, el argumento `method=''GCV.Cp''` indica la elección del parámetro de suavizado se realice mediante el método de validación cruzada ordinaria explicado en (2.8). El `summary` del modelo es el siguiente:

```
> summary(mod_gam)

Family: binomial
Link function: logit

Formula:
y ~ s(x, k = -1, bs = "ps")

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.1885     0.2062   0.914    0.361

Approximate significance of smooth terms:
       edf Ref.df Chi.sq p-value
s(x) 6.486   7.19  102.4  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.624   Deviance explained =   55%
UBRE = -0.32731  Scale est. = 1         n = 300
```

En esta salida se diferencian dos partes: la estimación de la parte paramétrica del modelo (β_0) y de la parte no paramétrica (f).

En la parte paramétrica, podemos ver que la estimación de β_0 es $\hat{\beta}_0 = 0.1885$. La última columna contiene el p-valor del contraste de significación del intercepto. Si fijamos un nivel de significación $\alpha = 0.05$, el p-valor de 0.361 indica que el β_0 no es significativo.

La tabla de la parte no paramétrica contiene datos sobre la estimación de la función f. La primera columna, *edf*, hace referencia a los grados de libertad necesarios en la estimación de la función, definidos en el segundo capítulo como (2.7). Representa una medida de complejidad de la función ajustada: cuanto mayor sea este valor, más oscilante será la curva. Un valor mayor que uno sugiere la necesidad de un modelo no lineal. En este caso, han utilizado 6.486 grados de libertad. En la Figura 2.7 aparece representada gráficamente la estimación de esta función. Podemos observar como se trata de una curva *suave* aunque con bastantes oscilaciones y cambios de pendiente. Junto a ella, las líneas discontinuas indican las bandas de confianza del 95 % para la estimación de la curva.

La última columna se corresponde con el p-valor de un contraste de significación de la función f sobre la variable respuesta. Para $\alpha = 0.05$, la función es significativamente distinta de cero. Es decir, la variable explicativa ejerce efecto en los valores de la variable respuesta.

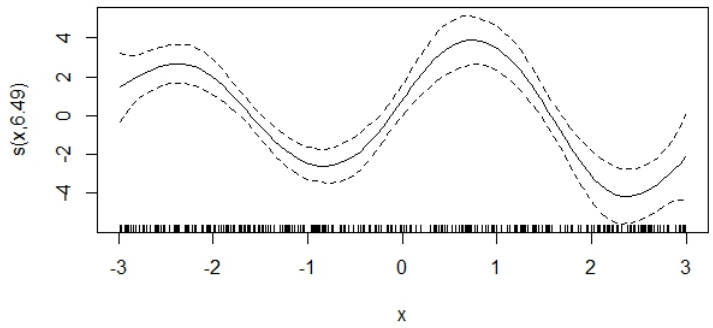
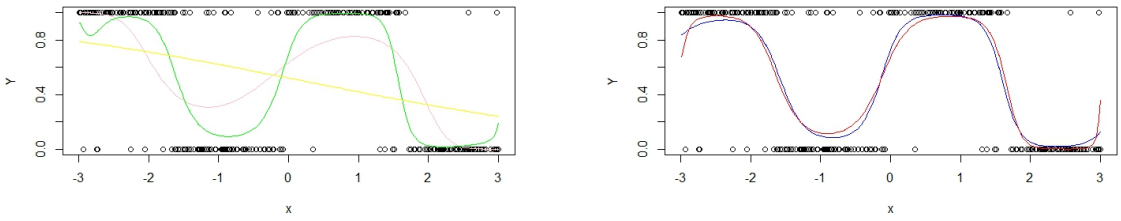


Figura 2.7: Representación gráfica de la estimación de la función *suave* del modelo (2.11).

Los valores de `R-sq.(adj)` y `Deviance explained` proporcionan información sobre la calidad del ajuste. A lo largo de este ejemplo y en el siguiente capítulo, debido a que trabajaremos con variables respuesta generalizadas, utilizaremos la *deviance* explicada (definida en 1.23) como medida de bondad de ajuste de los modelos. En este caso, la *deviance* explicada es de 0.55. Esto significa que el 55 % de la varianza del modelo nulo es explicada por el modelo (2.11).

Para ilustrar la necesidad contar con modelos aditivos generalizados, vamos a comparar el modelo (2.11) con diferentes modelos explicados en el capítulo uno. Todos los modelos considerados a partir de ahora serán modelos "generalizados", pero se omitirá este término por simplicidad a la hora de referirnos a ellos. Para la elección de los modelos con los que comparar (2.11) se ha tenido en cuenta los grados de libertad de las funciones. Como hemos visto, para la estimación de la función f de (2.11) se han utilizado 6.486 grados de libertad. Si tenemos en cuenta el intercepto, podemos considerar que para la estimación del modelo aditivo se han utilizado aproximadamente

siete grados de libertad. Por ello, para comparar este modelo se han ajustado cuatro modelos distintos con más grados de libertad, menos grados de libertad y aproximadamente igual número de grados de libertad. En la Figura 2.8 aparecen representados un modelo lineal (dos grados de libertad), un modelo polinómico de grado tres (cuatro grados de libertad), un modelo polinómico de grado doce (trece grados de libertad) y un modelo polinómico de grado seis (siete grados de libertad). A simple vista, en la Figura 2.8b podemos ver que los ajustes mediante el modelo aditivo y el modelo polinómico de grado seis son muy parecidos, diferenciándose un poco más en los valores extremos de la variable explicativa.



(a) (b)

Figura 2.8: Diagrama de dispersión del modelo (1.28) junto con las curvas para diferentes ajustes. A la izquierda, el modelo lineal generalizado (en amarillo), polinómico de grado tres (en rosa) y polinómico de grado doce (en verde). A la derecha, el modelo polinómico de grado seis (en rojo) y aditivo (en azul).

Para comparar analíticamente estos modelos entre sí, vamos a hacer uso de la *deviance* explicada, que se puede consultar en el Cuadro 2.1. Los valores de la *deviance* explicada para el modelo lineal (D_e=0.0854) y el modelo polinómico de grado tres (D_e=0.3380) son muy bajos. Esto indica un mal ajuste, que ya se podía apreciar en la Figura 2.8, debido a que resultan demasiado restrictivos para ajustar de forma precisa estos datos. Sin embargo, los siguientes tres modelos tienen valores muy próximos de D_e. Esta situación podría llevar a pensar que

Modelo	*Deviance* Explicada (D_e)
Lineal	0.0854
Polinómico grado 3	0.3380
Polinómico grado 6	0.5300
Polinómico grado 12	0.5545
Aditivo	0.5496

Cuadro 2.1: *Deviance* explicada para diferentes ajustes del modelo (1.28).

el planteamiento de un modelo aditivo no supone ninguna ventaja con respecto a un modelo polinómico para un grado de polinomio adecuado. No obstante, los modelos aditivos eliminan la necesidad de tener que elegir el grado de polinomio adecuado, lo no supone una tarea sencilla. Un polinomio que no cuente con los grados de libertad suficientes para estimar la función podría producir un subajuste de los datos (como sucede en ese caso con el modelo polinómico de grado tres), mientras que un polinomio con demasiados grados de libertad daría lugar a un sobreajuste de los datos y a un modelo demasiado complejo (como sucede con el modelo polinómico de grado doce). Además, la estimación mediante splines de regresión permite ajustar mediante splines de grado tres (que no son más que funciones polinómicas de grado tres a trozos con ciertas condiciones de regularidad) un modelo muy similar al determinado por una curva de grado seis, que resulta mucho más compleja.

Capítulo 3

Aplicación a datos reales

El objetivo principal de este capítulo es ilustrar los resultados sobre modelos aditivos generalizados vistos anteriormente a través de un ejemplo con datos reales. Para ello, utilizaremos una base de datos disponible en Wolberg et al. (1995) que recoge información acerca del cáncer de pecho. En primer lugar, definiremos las variables que contiene y a continuación, haremos un análisis exploratorio con el propósito de conocer su distribución y detectar las correlaciones existentes entre ellas. Por último, proseguiremos con el ajuste de distintos modelos que nos ayuden a explicar la probabilidad de que un tumor sea maligno haciendo uso de las variables contenidas en la base de datos.

3.1. Descripción de la base de datos

Como explica Street et al. (1993), en este estudio se miden diez características de tamaño, forma y textura de núcleos de células procedentes de tumores de pecho. La forma de estos núcleos se delimita mediante técnicas de procesamiento de imágenes. Para ello, se usan curvas conocidas como *snakes* que trazan de forma precisa la silueta de los núcleos y permiten digitalizar el análisis de sus características, lo que hace que sea más exacto que un simple análisis visual de la imagen. En cada uno de los *snakes* se seleccionan una serie de puntos, que llamaremos puntos *snake*, que facilitarán la medida de las características. En la Figura 3.1 aparecen representadas estas curvas *snake* para los núcleos celulares existentes en una muestra de tejido del tumor. Las medidas individuales de cada núcleo celular son las siguientes:

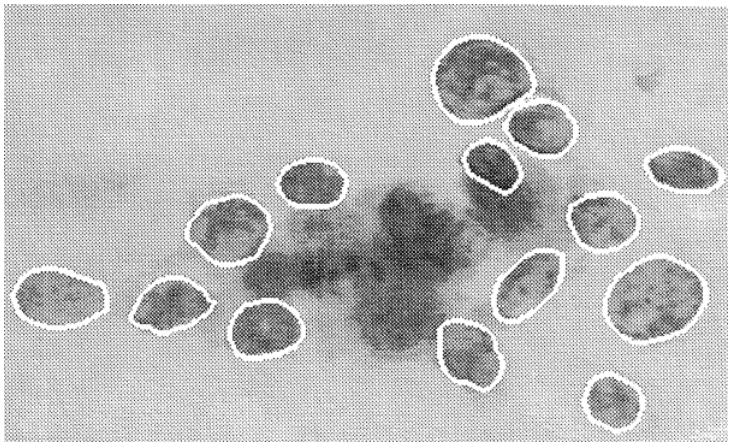


Figura 3.1: Núcleos celulares presentes en una muestra de tejido tumoral delimitados por curvas *snakes* (Street et al., 1993).

1. Radius (radio): media de las distancias desde el centroide del núcleo a cada uno de los puntos *snake*.

2. Perimeter (perímetro): distancia total entre los puntos *snake*.

3. Area (área): medida del tamaño del núcleo celular que se calcula sumando el número de píxeles que contiene el *snake* y un medio del número de píxeles del perímetro.

4. Compactness (compacidad): medida de compacidad del núcleo celular calculada como $Perimeter^2/Area$.

5. Smoothness (suavidad): medida de suavidad del contorno del núcleo que se obtiene como la diferencia entre la longitud del radio en un punto *snake* y la longitud media de los radios adyacentes (ver Figura 3.2a).

6. Concavity (concavidad): medida que refleja el número y forma de las secciones cóncavas del núcleo celular. Se dibujan cuerdas entre puntos *snake* no adyacentes y se calcula la distancia al límite real de cada una de las células. De esta forma, se mide el tamaño de las hendiduras del contorno (ver Figura 3.2b).

7. Concave points (puntos cóncavos): número de secciones cóncavas del contorno nuclear.

8. Simmetry (simetría): medida de simetría del núcleo celular. Se obtiene dividiendo el núcleo por un eje mayor que pase por su centro y trazando líneas perpendiculares a este eje. Luego, se calcula la diferencia entre la distancias de estas líneas perpendiculares a un lado y al otro del eje (ver Figura 3.2c).

9. Fractal dimension (dimensión fractal): medida que refleja la irregularidad del contorno del núcleo celular. Para su cálculo, se mide el perímetro del núcleo celular utilizando polígonos que aproximen su forma. A mayor número de lados de los polígonos, mejor sera la aproximación del perímetro. Se representan estos valores en orden descendente en una escala logarítmica y se calcula su pendiente. La dimensión fractal es el opuesto de este valor.

10. Texture (textura): medida de textura del núcleo celular que se obtiene a través de la variación de la escala de grises de los píxeles que forman el núcleo.

Estas diez características se podrían organizar en tres bloques: medidas de tamaño (radio, perímetro y área), medidas de relieve (textura y compacidad) y medidas de forma (suavidad, concavidad, puntos cóncavos, simetría y dimensión fractal).

(a) Suavidad (b) Concavidad (c) Simetría

Figura 3.2: Representación gráfica de la medición de las características de suavidad, concavidad y simetría de los núcleos celulares (Street et al., 1993).

Se analizan 569 imágenes y se calcula (para cada una de las diez características anteriores) la media, la desviación típica y el valor extremo (más alto) de los núcleos celulares que aparecen en cada una de las imágenes. Denotaremos estos valores por `característica_mean`, `característica_se` y `característica_worst` respectivamente, donde `característica` representa el nombre de cada una de las diez características explicadas anteriormente. Por tanto, la base de datos recoge información de treinta variables explicativas procedentes de 569 observaciones.

Como variable respuesta consideraremos la variable llamada `diagnosis`. En ella se almacena información sobre si cada uno de los 569 tumores analizados es benigno (`B`) o maligno (`M`). Para poder incluir esta variable en nuestros modelos, la convertimos en una variable binaria que tome los valores 0 (el tumor es benigno) y 1 (el tumor es maligno).

Debido a que la presencia de un único núcleo celular maligno provoca que el tumor sea considerado como maligno, las variables explicativas que almacenan información sobre los valores más altos de cada imagen (las denotadas como `característica_worst`) serán más relevantes en

el análisis de la naturaleza de los tumores. Por este motivo, serán las únicas que consideraremos en este trabajo.

3.2. Análisis exploratorio de los datos

En primer lugar, realizamos un análisis exploratorio de los datos que nos puede servir para entender sus características y detectar relaciones entre las distintas variables. En la siguiente tabla mostramos las principales medidas descriptivas de cada una de las variables:

Variable	Media	Desviación típica	Mediana	Mínimo	Máximo
Radio	16.2692	4.8332	14.9700	7.9300	36.0400
Textura	25.6772	6.1463	25.4100	12.0200	49.5400
Perímetro	107.2612	33.6025	97.6600	50.4100	251.2000
Área	880.5831	569.3570	686.5000	185.2000	4254.0000
Suavidad	0.1324	0.0228	0.1313	0.0712	0.2226
Compacidad	0.2543	0.1573	0.2119	0.0273	1.0580
Concavidad	0.2722	0.2086	0.2267	0.0000	1.2520
Puntos cóncavos	0.1146	0.0657	0.0999	0.0000	0.2910
Simetría	0.2901	0.0619	0.2822	0.1565	0.6638
Dimensión fractal	0.0839	0.0181	0.0800	0.0550	0.2075

Cuadro 3.1: Medidas descriptivas de las variables explicativas.

Por otra parte, de entre los 569 tumores analizados, 357 son benignos mientras que 212 de ellos son malignos.

Puesto que existen varias variables que miden características entre las que podría haber una relación, calculamos las correlaciones entre todos los pares de variables. Estas correlaciones aparecen representadas en el mapa de calor de la Figura 3.3. Como todas las correlaciones son positivas, se ha considerado una escala que va de cero a uno. Podemos ver que, efectivamente, las variables relacionadas con el tamaño de los núcleos presentan una correlación positiva muy fuerte. Por ejemplo, el radio y el perímetro (correlación casi perfecta de 0.99), el radio y el área (correlación de 0.98) o el perímetro y el área (correlación de 0.98). Además, también existen correlaciones muy importantes entre la compacidad y la concavidad (0.89) o la concavidad y los puntos cóncavos (0.86). Todas estas relaciones eran previsibles debido a la descripción de las variables expuesta anteriormente.

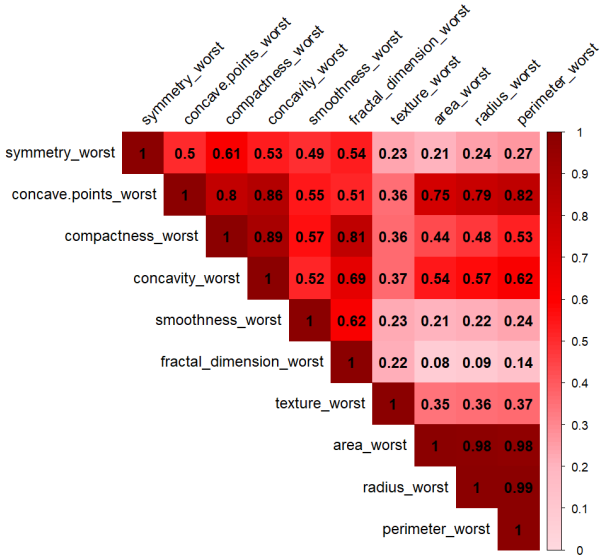


Figura 3.3: Mapa de calor para las correlaciones entre las variables explicativas

Por otro lado, en la Figura 3.4 aparecen representados los histogramas de frecuencias absolutas de la naturaleza del tumor para las variables consideradas y en la Figura 3.5, los gráficos de densidad para los tumores benignos y malignos para cada una de las características. Podemos ver que todas las variables cumplen que para valores más altos de cada una de las características, aumenta el número de tumores malignos, llegando incluso a desaparecer los tumores benignos.

3.3. Ajuste de los modelos

En esta sección utilizaremos la base de datos descrita anteriormente con el propósito de ajustar modelos que nos ayuden a ilustrar los resultados vistos a lo largo del trabajo. Intentaremos predecir la probabilidad de que un tumor sea maligno a través de las características de tamaño, forma y relieve explicadas en la primera sección de este capítulo. Debido a la gran cantidad de variables explicativas y técnicas de suavizado estudiadas, se podrían ajustar multitud de modelos distintos. Sin embargo, más allá de encontrar el modelo que mejor explique los datos, nos centraremos en ejemplificar e implementar en  algunas de las técnicas explicadas en el capítulo dos, además de motivar el uso de los modelos aditivos generalizados. La estimación de las funciones *suaves* consideradas en este capítulo se llevará a cabo mediante P-splines (con una base de B-splines de grado tres y penalizaciones basadas en diferencias de orden dos) y el parámetro de suavizado se elegirá mediante el método de validación cruzada generalizada.

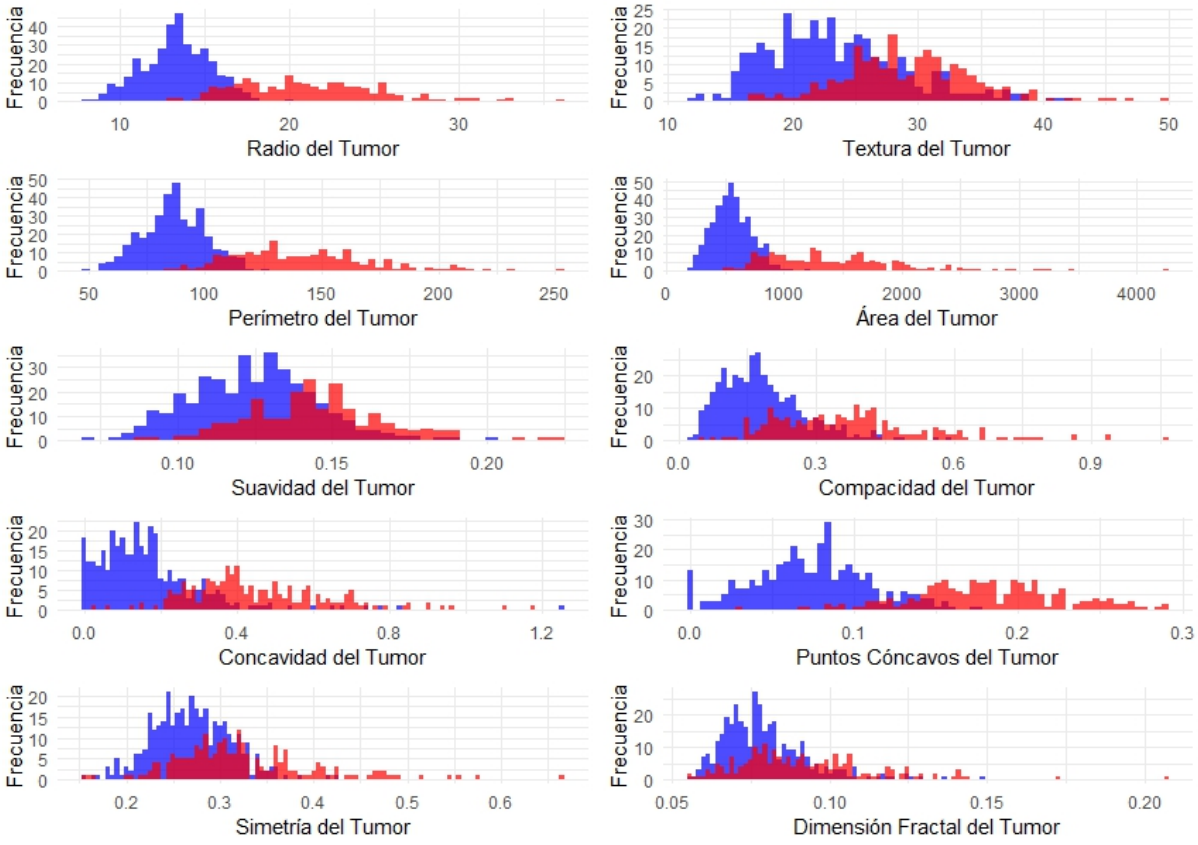


Figura 3.4: Histogramas de frecuencias absolutas de cada una de las variables. Las barras azules indican el número de tumores benignos en cada intervalo de la característica representada en el histograma. Las barras rojas muestran el número de tumores malignos en cada intervalo de la característica. Las barras magentas aparecen en los intervalos donde coexisten tumores benignos y malignos, es decir, que en rangos específicos de la característica, se observan ambos tipos de tumores.

En primer lugar, ajustaremos un modelo aditivo generalizado muy sencillo. En él, se considerará como única variable explicativa x_1 la variable `texture_worst`. Esta se ha elegido tras analizar todos los modelos aditivos generalizados ajustados con una sola variable explicativa y observar que la relación entre `texture_worst` y probabilidad de éxito de la variable respuesta está lejos de ser lineal. Por este motivo, resulta interesante para ilustrar técnicas de suavizado. El modelo se podría formular de la siguiente forma:

$$\log\left(\frac{\pi(x_1)}{1-\pi(x_1)}\right) = \beta_0 + f_1(x_1) \tag{3.1}$$

donde $\pi(x_1)$ representa la probabilidad de que el tumor sea maligno condicionada al valor de la variable explicativa, β_0 es el intercepto [1] (y en este caso, la única parte paramétrica del modelo)

[1] En esta sección se denotarán como β_l los coeficientes de la parte paramétrica del modelo.

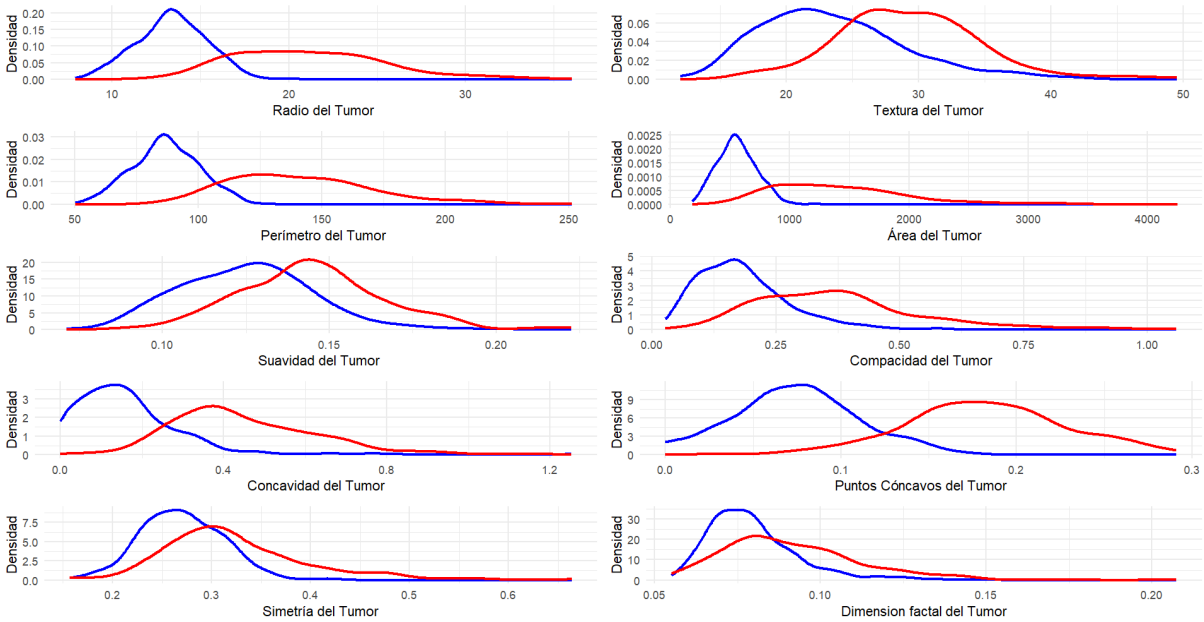


Figura 3.5: Gráficos de densidad de cada una de las variables. En azul, la curva de densidad para los tumores benignos. En rojo, la curva de densidad para los tumores malignos.

y f_1 es una función *suave* de la variable explicativa considerada. La salida del `summary` se resume en los Cuadros 3.2 y 3.3.

	Estimate	Std. Error	z value	Pr(>|z|)
Intercepto	-0.7591	0.1160	-6.547	5.89e-11

Cuadro 3.2: Estimación de los coeficientes paramétricos del modelo (3.1).

Funciones *suaves*	edf	Ref.df	Chi.sq	p-value
s(`texture_worst`)	3.583	4.3	101.6	<2e-16

Cuadro 3.3: Estimación de los términos suaves no paramétricos del modelo (3.1).

En la tabla de la parte paramétrica (Cuadro 3.2), podemos ver que la estimación de β_0 es $\hat{\beta}_0 = -0.7591$. Si fijamos un nivel de significación $\alpha = 0.05$, el p-valor de la última columna indica que el intercepto es significativamente distinto de cero.

La tabla de la parte no paramétrica (Cuadro 3.3), muestra que para la estimación de la función f_1 se utilizaron 3.583 grados de libertad. El contraste de significación de la función f_1

sobre la variable explicativa tiene un p-valor menor que 2e-16, lo que indica que para un nivel de confianza del 95 %, la función es distinta de cero. Es decir, la variable `texture_worst` ejerce efecto en los valores de la variable respuesta. En la Figura 3.6 aparece representada la función f_1. Como indicaba el valor de sus grados de libertad (por ser mayor que uno), se trata de una función no lineal. Los segmentos del eje de abscisas indican los valores de `texture_worst`. Puede observarse que en aquellos intervalos que cuentan con menos valores de la variable explicativa las bandas de confianza son más amplias.

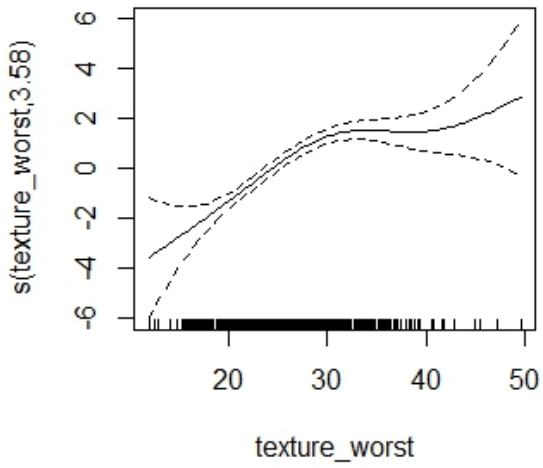


Figura 3.6: Representación gráfica de la estimación de la función suave del modelo (3.1).

El valor de la *deviance* explicada es de 0.198, el cual es bastante bajo e indica un mal ajuste de los datos. Para intentar mejorar este modelo, vamos a introducir otra variable explicativa. Recordemos que las variables medían características de tamaño, forma y relieve de núcleos celulares. El modelo (3.1) incluye una sola variable explicativa relativa al relieve del núcleo (`texture_worst`). Vamos a añadir al modelo anterior una variable explicativa, x_2, relativa al tamaño, como `radius_worst`. El modelo ajustado tendrá la siguiente forma:

$$\log\left(\frac{\pi(x_1,x_2)}{1-\pi(x_1,x_2)}\right) = \beta_0 + f_1(x_1) + f_2(x_2) \tag{3.2}$$

donde f_2 es una función *suave* de la variable `radius_worst`. La salida del `summary` del modelo (3.2) se recoge en los Cuadros 3.4 y 3.5.

	Estimate	Std. Error	z value	Pr(>|z|)
Intercepto	-0.6535	0.2158	-3.028	0.00246

Cuadro 3.4: Estimación de los coeficientes paramétricos del modelo (3.2).

Funciones *suaves*	edf	Ref.df	Chi.sq	p-value
s(`texture_worst`)	2.078	2.548	38.47	<2e-16
s(`radius_worst`)	1.000	1.001	76.41	<2e-16

Cuadro 3.5: Estimación de los términos suaves no paramétricos del modelo (3.2).

En ellos, se muestra que tanto el coeficiente de la parte paramétrica como los efectos *suaves* de las variables explicativas son significativos. Sin embargo, si nos fijamos en los grados de libertad, para la estimación de la función f_2 se está utilizando un único grado de libertad. Esto significa que se puede considerar que el efecto de la variable `radius_worst` sobre la logit de probabilidad de éxito de variable respuesta es lineal, como se puede ver en la gráfica de la derecha de la Figura 3.7. Por otra parte, el valor de la *deviance* explicada es 0.774. Comparando este valor con el obtenido para el modelo (3.1), podemos ver como la introducción de la variable `radius_worst` ha mejorado considerablemente el modelo.

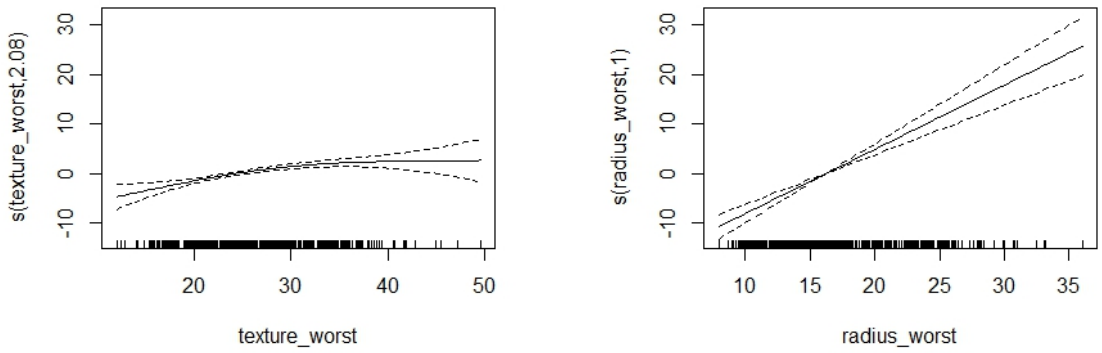


Figura 3.7: Funciones *suaves* de las variables explicativas del modelo (3.2).

Con el fin de simplificar el modelo (3.2) y teniendo en cuenta que el efecto lineal de `radius_worst`, vamos a plantear un nuevo modelo en el que incluyamos esta variable en la parte paramétrica del modelo:

$$\log\left(\frac{\pi(x_1,x_2)}{1-\pi(x_1,x_2)}\right) = \beta_0 + f_1(x_1) + \beta_2 x_2 \tag{3.3}$$

donde β_2 es el coeficiente de la variable `radius_worst`. Resumimos la salida del `summary` en el Cuadro 3.6 y 3.7. Como podemos comprobar comparando los Cuadros 3.5 y 3.7, la estimación de f_1 es la misma en ambos modelos. Para el modelo (3.3) obtenemos un valor para la *deviance* escalada de 0.774, igual al del modelo (3.2).

	Estimate	Std. Error	z value	Pr(>|z|)
Intercepto	-21.8249	2.4123	-9.047	<2e-16
radius_worst	1.3013	0.1488	8.745	<2e-16

Cuadro 3.6: Estimación de los coeficientes paramétricos del modelo (3.3).

Funciones suaves	edf	Ref.df	Chi.sq	p-value
s(`texture_worst`)	2.078	2.548	38.47	<2e-16
s(`radius_worst`)	1.000	1.001	76.41	<2e-16

Cuadro 3.7: Estimación de los términos suaves no paramétricos del modelo (3.3).

Con el fin de seguir mejorando los resultados obtenidos, introducimos en (3.3) una variable explicativa, x_3, que describa la forma de los núcleos celulares, como `concave.points_worst`. De esta manera, estaremos considerando una variable explicativa por cada uno de los tres tipos de características (relieve, tamaño y forma). El planteamiento del modelo es el siguiente:

$$\log\left(\frac{\pi(x_1,x_2,x_3)}{1-\pi(x_1,x_2,x_3)}\right) = \beta_0 + f_1(x_1) + \beta_2 x_2 + f_3(x_3) \tag{3.4}$$

donde f_3 es una función *suave* de la variable explicativa `concave.points_worst`. El `summary` de este modelo se recoge en el Cuadro 3.8 y 3.9.

	Estimate	Std. Error	z value	Pr(>|z|)
Intercepto	-20.258	3.203	-6.325	2.54e-10
radius_worst	1.223	0.197	6.208	5.37e-10

Cuadro 3.8: Estimación de los coeficientes paramétricos del modelo (3.4).

Como se puede observar, todos los coeficientes de la parte paramétrica y efectos suaves son significativos para un nivel de significación del 5 %. En la Figura 3.8 aparecen representadas las componentes suaves del modelo. Ninguna de las dos variables (`texture_worst` y

Funciones suaves	edf	Ref.df	Chi.sq	p-value
s(texture_worst)	3.157	3.733	23.71	8.53e-05
s(concave.points_worst)	2.144	2.669	26.95	5.34e-06

Cuadro 3.9: Estimación de los términos suaves no paramétricos del modelo (3.4).

`concave.points_worst`) ejerce un efecto lineal sobre la probabilidad de éxito la variable respuesta, ya que se han estimado con 3.157 y 2.144 grados de libertad respectivamente. Este modelo cuenta con un valor de *deviance* explicada de 0.88. Por ser próximo a uno, indica que el modelo (3.4) logra explicar el 88 % de la variabilidad presente en los datos y que por tanto, se trata de un modelo con una buena calidad de ajuste.

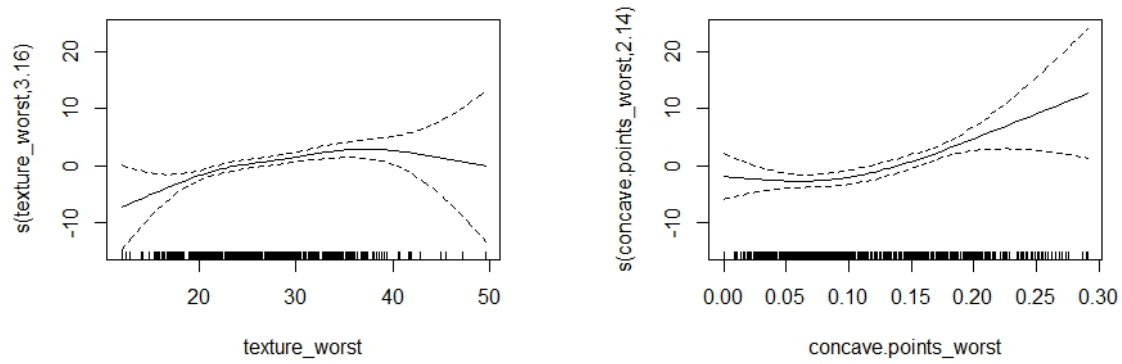


Figura 3.8: Funciones *suaves* de las variables explicativas del modelo (3.4).

A lo largo de esta sección hemos recurrido a la *deviance* explicada para evaluar la calidad del ajuste. Sin embargo, es natural pensar que estos modelos, además de para explicar los datos, podrían ser útiles para hacer predicciones sobre la naturaleza de los tumores. Que un modelo se ajuste bien a unos datos no siempre significa que sea capaz de predecir de forma correcta nuevas observaciones. Por ejemplo, las medidas de bondad de ajuste para un modelo que interpole los datos serían muy buenas. Sin embargo, tendría escasa capacidad de predicción para nuevos valores de las variables explicativas.

Para evaluar la capacidad predictiva de los modelos (3.1), (3.2), (3.3) y (3.4), dividimos aleatoriamente las 569 observaciones de la muestra en dos grupos: el conjunto de entrenamiento (469 observaciones) y el conjunto test (100 observaciones). Ajustamos los modelos usando el conjunto de entrenamiento y usamos el conjunto test para cuantificar su capacidad predictiva. Para ello, comparamos las observaciones del conjunto test con las predicciones obtenidas por cada uno de los cuatro modelos para estas observaciones. El porcentaje de acierto de las predicciones

se recoge en el Cuadro (3.10). Como podemos ver, el modelo (3.1) es el que cuenta con el porcentaje considerablemente más bajo. El resto de los modelos tienen un porcentaje de acierto igual o mayor al 90 %, siendo modelo (3.4) es el que tiene mayor tasa de acierto. Como habíamos visto anteriormente, también es el que cuenta con mayor *deviance* explicada. Por tanto, esta prueba ilustra que los modelos aditivos no solo mejoran la capacidad explicativa del modelo, sino que permite hacer mejores predicciones.

Modelo	Porcentaje de acierto (%)
Modelo (3.1)	66
Modelo (3.2)	90
Modelo (3.3)	93
Modelo (3.4)	96

Cuadro 3.10: Porcentaje de acierto de las predicciones realizadas por los cuatro modelos sobre el conjunto test.

3.4. Conclusiones

A través de los modelos de la sección anterior hemos ilustrado el manejo básico de la función `gam` de R. Además, hemos conseguido ajustar un modelo, el (3.4), capaz de explicar de forma bastante precisa la probabilidad de que un tumor sea maligno en función de tres de sus características: textura, radio y puntos cóncavos. Como hemos comentado, se trata de un ejemplo muy sencillo en que el principal objetivo era ilustrar el uso de los modelos aditivos generalizados. Aunque no se ha tratado en este trabajo, estos modelos también permiten la introducción de variables categóricas e interacciones entre las variables. A pesar de su simplicidad, este ejemplo deja entrever la importancia de contar con modelos capaces de adaptarse a datos procedentes del mundo real para ayudarnos a entenderlos y predecirlos.

Anexo I

Código de 

A continuación proporcionamos el código utilizado en el trabajo.

Código para la simulación de datos y ajuste de los modelos de la Figura 1.1:

```r
# Simulación de los datos
set.seed(20518)
n <- 300
x_orig <- runif(n,-2,2)
x <- sort(x_orig)
mod_x <- 0.5+4*x
p_x <- 1/(1+exp(-mod_x))
summary(p_x)
y <- numeric(n)
for (k in 1:n){
  y[k] <- rbinom(n=1,size=1,prob=p_x[k])
}

# Modelo lineal simple obviando que la respuesta es binaria
mod_lineal <- lm(y ~ x)
plot(x,y, ylab="Y",ylim=c(-0.2, 1.2))
abline(mod_lineal,lwd=1.5,col="blue")

# Modelo linealizable polinómico de orden tres obviando que la respuesta es binaria
x2 <- x^2; x3 <- x^3
mod_pol3 <- lm(y ~ x + x2 + x3)
```

```
22 points(x, fitted(mod_pol3), type="l",lwd=1.5, col="green")
23
24 # Modelo linealizable polinómico de orden diez obviando que la respuesta es binaria
25 x4 <- x^4; x5 <- x^5; x6 <- x^6; x7 <- x^7; x8 <- x^8; x9 <- x^9; x10 <- x^10
26 mod_pol10 <- lm(y ~ x + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10)
27 points(x, fitted(mod_pol10), type="l", col="red",lwd=1.5)
28
```

Código para la simulación de datos y ajuste de los modelos de la Figura 1.2:

```
1  #Simulacion de los datos
2  x <- 0.01*(-400:400)
3  n <- length(x)
4  e<- numeric(n)
5  set.seed(2785)
6  for (k in 1:n){
7    e[k] <- rnorm(n=1,mean=0,sd=0.5)
8  }
9  y <- 0.5+cos(2*x)+exp(cos(x))+e
10 plot(x,y,ylab="Y")
11
12 # Ajuste de un modelo lineal
13 mod_lineal = lm(y ~ x)
14 abline(mod_lineal,col="blue")
15 summary(mod_lineal)
16
17 # Ajuste de modelos linealizables
18 # Modelo polinómico de orden 2
19 x2=x^2
20 mod_pol2=lm(y ~ x+x2)
21 summary(mod_pol2)
22 points(x,fitted(mod_pol2),col="red",type="l")
23
24 # Modelo polinómico de orden 3
25 x3=x^3
26 mod_pol3=lm(y ~ x+x2+x3)
27 points(x,fitted(mod_pol3),col="green",type="l")
```

```
# Modelo polinómico de orden 4
x4=x^4
mod_pol3=lm(y ~ x+x2+x3+x4)
summary(mod_pol4)
points(x,fitted(mod_pol3),col="red",type="l")

# Modelo polinómico de orden 8
x5=x^5;x6=x^6;x7=x^7;x8=x^8
mod_pol8=lm(y ~ x+x2+x3+x4+x5+x6+x7+x8)
summary(mod_pol8)
points(x,fitted(mod_pol8),col="yellow",type="l")

```

Código para la Figura 1.3:

```
# Binomial
n <- 40
p <- 0.4
x1 <- 0:n
densidad1 <- dbinom(x, size =n, prob = p)
plot(x1, densidad3, lwd = 2, pch = 16, col = "blue",
     xlab = "Valores", ylab = "Probabilidad")

# Poisson
x2 <- 0:15
lambda <- 5
densidad2 <- dpois(x2, lambda)
plot(x2, densidad2, lwd = 2, pch = 16, col = "blue",
     xlab = "Valores", ylab = "Probabilidad")

# Gamma
s <- 4
r <- 1
x3 <- seq(0, 10, length.out = 100)
densidads3 <- dgamma(x3, s, r)
plot(x3, densidades2, type = "l", lwd = 2, col = "blue",
     xlab = "Valores", ylab = "Densidad de probabilidad")
```

Código para la simulación de datos y ajuste de los modelos de la Figura 1.4:

```
#Simulacion de los datos
set.seed(20518)
n <- 300
x_orig <- runif(n,-3,3)
x <- sort(x_orig)
mod_x <- 0.3+4*sin(2*x)+cos(x)
p_x <- 1/(1+exp(-mod_x))
y <- numeric(n)
for (k in 1:n){
  y[k] <- rbinom(n=1,size=1,prob=p_x[k])
}

# Modelo lineal generalizado
mod_logit <- glm(y ~ x, family=binomial(link="logit"))
summary(mod_logit)

# Modelo linealizable polinómico de orden dos generalizado
x2=x^2
mod_logit_pol2 <- glm(y ~ x + x2, family=binomial(link="logit"))
summary(mod_logit_pol2)


# Modelo linealizable polinómico de orden cuatro generalizado
x3=x^3;x4=x^4
mod_logit_pol4 <- glm(y ~ x+x2+x3+x4, family=binomial(link="logit"))
summary(mod_logit_pol4)

# Modelo linealizable polinómico de orden diez generalizado
x5=x^5; x6=x^6; x7=x^7; x8=x^8; x9=x^9; x10=x^10
mod_logit_pol10 <- glm(y ~ x + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10, family=binomial(l
summary(mod_logit_pol10)

#Representacion gráfica
par(mfrow=c(1,3))
plot(x,mod_x,xlab="x", ylab = "f(x)")
```

```
points(x, predict(mod_logit),  col="blue",type="l", lwd=1.5)
points(x, predict(mod_logit_pol2), col="green", type="l", lwd=1.5)
points(x, predict(mod_logit_pol10), col="yellow", type="l",lwd=1.5)
points(x, predict(mod_logit_pol4), col="red", type="l",lwd=1.5)

plot(x,p_x,ylab = ~pi(x))
points(x, fitted(mod_logit),col="blue", type="l", lwd=1.5)
points(x, fitted(mod_logit_pol2), col="green", type="l", lwd=1.5)
points(x, fitted(mod_logit_pol4), col="red", type="l",lwd=1.5)
points(x, fitted(mod_logit_pol10), col="yellow", type="l",lwd=1.5)

plot(x,y)
points(x, fitted(mod_logit),col="blue", type="l", lwd=1.5)
points(x, fitted(mod_logit_pol2), col="green", type="l", lwd=1.5)
points(x, fitted(mod_logit_pol4), col="red", type="l",lwd=1.5)
points(x, fitted(mod_logit_pol10), col="yellow", type="l",lwd=1.5)
```

Código para la Figura 1.5:

```
# Simulación 1
n1 <- 500
x1 <- seq(-1,2,length.out=n1)
y1 <- numeric(n1)
y1[which(x1 <= -0.5)] <- -10*x1[which(x1 <= -0.5)]- 7+
rnorm(length(which(x1 <= -0.5)),sd=0.3)
y1[which(x1 <= 0.5 & x1 > -0.5)] <- 3*x1[which(x1 <= 0.5 & x1 > -0.5)] -1.5 +
rnorm(length(which(x1 <= 0.5 & x1 > -0.5)),sd=0.3)
y1[which(x1 > 0.5 & x1 < 2)] <- -cos(10*x1[which(x1 > 0.5 & x1 < 2)]) +
rnorm(length(which(x1 > 0.5 & x1 < 2)),sd=0.3)
plot(x1,y1,ylab="Y",xlab="x")

# Simulación 2
x2 <- 0.2*(300:600)
n2 <- length(x2)
e2<- numeric(n2)
for (k in 1:n2){
  e2[k] <- rnorm(n=1,mean=0,sd=0.2)
```

```
}
y2<-exp(sin(x2))+e2
plot(x2,y2,ylab="Y",xlab="x")
```

Código para la Figura 2.1:

```
x= seq(from = 0, to = 5, length.out = 500)
nodos<-c(1,1.5,3,4,4.5)

# Spline de grado uno:
# Funciones básicas:
x11 <- rep(1, length(x))
x12 <- x
x13 <- pmax(0, x - 1)
x14 <- pmax(0, x - 1.5)
x15 <- pmax(0, x - 3)
x16 <- pmax(0, x - 4)
x17 <- pmax(0, x - 4.5)

# Representación gráfica:
y1=-2*x11-3*x12+3*x13+0.1*x14-4*x15+2*x16-2*x17 #función spline
plot(x,y1,type = "l",col="red", ylab="Spline de grado 1")
abline(v = 1, lty = 2, lwd = 1)
abline(v = 1.5, lty = 2, lwd = 1)
abline(v = 3, lty = 2, lwd = 1)
abline(v = 4, lty = 2, lwd = 1)
abline(v = 4.5, lty = 2, lwd = 1)

# Spline de grado dos:
# Funciones básicas:
x21 <- rep(1, length(x))
x22 <- x
x23 <- x^2
x24 <- pmax(0, x - 1)^2
x25 <- pmax(0, x - 1.5)^2
x26 <- pmax(0, x - 3)^2
x27 <- pmax(0, x - 4)^2
```

```
x28 <- pmax(0, x - 4.5)^2

# Representación gráfica:
y2= -2*x21 + 0.5*x22 + 5*x23 + 10*x24 - 20*x25 + 30*x26 - 70*x27 + 100*x28
plot(x,y2,type = "l",col="red", ylab="Spline de grado 2")
abline(v = 1, lty = 2, lwd = 1)
abline(v = 1.5, lty = 2, lwd = 1)
abline(v = 3, lty = 2, lwd = 1)
abline(v = 4, lty = 2, lwd = 1)
abline(v = 4.5, lty = 2, lwd = 1)
# Spline de grado tres:
# Funciones básicas:
x31 <- rep(1, length(x))
x32 <- x
x33 <- x^2
x34 <- x^3
x35 <- pmax(0, x - 1)^3
x36 <- pmax(0, x - 1.5)^3
x37 <- pmax(0, x - 3)^3
x38 <- pmax(0, x - 4)^3
x39 <- pmax(0, x - 4.5)^3

# Representación gráfica:
y3 <- 10*x31 + 8*x32 + 25*x33 - 7*x34 + 10*x35 - 30*x36 + 10*x37 - 45*x38 + 50*x39
plot(x,y3,type = "l",col="red",ylab="Spline de grado 3")
abline(v = 1, lty = 2, lwd = 1)
abline(v = 1.5, lty = 2, lwd = 1)
abline(v = 3, lty = 2, lwd = 1)
abline(v = 4, lty = 2, lwd = 1)
abline(v = 4.5, lty = 2, lwd = 1)
```

Código para el ajuste del modelo 1.4 de la Figura 2.2:

```
modelo1<-smooth.spline(x,y,lambda=10,all.knots = TRUE)
modelo2<-smooth.spline(x,y,lambda=0.001,all.knots = TRUE)
modelo3<-smooth.spline(x,y,lambda=0.00001,all.knots = TRUE)

```

```
#Representación gráfica
plot(x,y,ylab="Y")
points(x,fitted.values(modelo1),type="l",col="red",lwd=2)
plot(x,y,ylab="Y")
points(x,fitted.values(modelo2),type="l",col="red",lwd=2)
plot(x,y,ylab="Y")
points(x,fitted.values(modelo3),type="l",col="red",lwd=2)
```

Código para la Figura 2.3:

```
x= seq(from = 0, to = 1, length.out = 100)
nodos<-c(0.2,0.7)

# Funciones básicas de polinomios truncados de grado uno:
x11 <- rep(1, length(x))
x12 <- x
x13 <- pmax(0, x - 0.2)
x14 <- pmax(0, x - 0.7)

# Representación gráfica:
plot(x,x11,type="l",col="red",lwd=1.5,ylim=c(-0.5,1.5),ylab="y")
abline(v = 0.2, lty = 2, lwd = 1)
abline(v = 0.7, lty = 2, lwd = 1)
points(x,x12,type = "l",col="green",lwd = 1.5)
points(x,x13,type = "l",col="blue",lwd = 1.5)
points(x,x14,type = "l",col="orange",lwd = 1.5)

# Funciones basicas de polinomios truncados de grado dos:
x21 <- rep(1, length(x))
x22 <- x
x23 <- x^2
x24 <- pmax(0, x - 0.2)^2
x25 <- pmax(0, x - 0.7)^2

# Representación gráfica:
plot(x,x21,type = "l",col="red",lwd = 1.5,ylim=c(-0.5,1.5),ylab="y")
abline(v = 0.2, lty = 2, lwd = 1)
```

```
abline(v = 0.7, lty = 2, lwd = 1)
points(x,x22,type = "l",col="green",lwd = 1.5)
points(x,x23,type = "l",col="blue",lwd = 1.5)
points(x,x24,type = "l",col="purple",lwd = 1.5)
points(x,x25,type = "l",col="orange",lwd = 1.5)
```

Código para la Figura 2.4:

```
x<- seq(-5, 5, length.out = 1000)
nodos <- -4:4

# Base de B-splines de grado uno
library("splines")
base1 <- bs(x, knots = nodos, degree = 1, intercept = FALSE, Boundary.knots = range(x))

# Algunas funciones básicas son:
b11 <- base1[, 2]
b12 <- base1[, 5]
b13 <- base1[, 6]
b14 <- base1[, 9]

# Representación de estas funciones básicas:
plot(x, b11, xlim = c(-5, 5), ylim = c(0, 1), type = "l", col = "red", lwd = 1.5,
     xlab = " ", ylab = "")
lines(x, b12, col = "blue", lwd = 1.5)
lines(x, b13, col = "green", lwd = 1.5)
lines(x, b14, col = "orange", lwd = 1.5)
axis(1, at = seq(-5, 5, by = 1))

# Base de B-splines de grado dos
base2 <- bs(x, knots = nodos, degree = 2, intercept = FALSE, Boundary.knots = range(x))
dim(base2)

# Algunas funciones básicas son:
b21 <- base2[, 2]
b22 <- base2[, 5]
b23 <- base2[, 6]
```

```
b24 <- base2[, 9]

# Representación de estas funciones básicas:
plot(x, b21, xlim = c(-5, 5), ylim = c(0, 1), type = "l", col = "red", lwd = 1.5,
     xlab = "x", ylab = "")
lines(x, b22, col = "blue", lwd = 1.5)
lines(x, b23, col = "green", lwd = 1.5)
lines(x, b24, col = "orange", lwd = 1.5)
axis(1, at = seq(-5, 5, by = 1))
```

Código para la Figura 2.5:

```
# Ajuste de los modelos usando bases de polinomios truncados:
# Con 4 nodos, funciones de la base:
nodos1=seq(-4, 4, length.out = 4)
x11 <- rep(1, length(x))
x12 <- x
x13 <- x^2
x14 <- x^3
x15 <- pmax(0, x-nodos1[1])^3
x16 <- pmax(0, x-nodos1[2])^3
x17 <- pmax(0, x-nodos1[3])^3
x18 <- pmax(0, x-nodos1[4])^3

# Ajuste del modelo:
modelopol1<-lm(y ~x11 +x12+x13+x14+x15+x16+x17+x18)
plot(x,y,ylab="Y")
points(x,fitted(modelopol1),col="red",type="l",lwd=2)

# Con 12 nodos, funciones de la base:
nodos2<-seq(-4, 4, length.out = 12)
x21 <- rep(1, length(x))
x22 <- x
x23 <- x^2
x24 <- x^3
x25 <- pmax(0, x -nodos1[1])^3
x26 <- pmax(0, x-nodos1[2])^3
```

```
x27 <- pmax(0, x-nodos1[3])^3
x28 <- pmax(0, x-nodos1[4])^3
x29<- pmax(0, x-nodos1[5])^3
x210<- pmax(0, x-nodos1[6])^3
x211<- pmax(0, x-nodos1[7])^3
x212<- pmax(0, x-nodos1[8])^3
x213<- pmax(0, x-nodos1[9])^3
x214<- pmax(0, x-nodos1[10])^3
x215<- pmax(0, x-nodos1[11])^3
x216<- pmax(0, x-nodos1[12])^3

# Ajuste del modelo:
modelopol2<-lm(y ~x21 +x22+x23+x24+x25+x26+x27+x28+x29+x210+x211+x212+x213+x214+x215+x216)
points(x,fitted(modelopol2),col="green",type="l",lwd=2)

# Ajuste de los modelos usando bases de B-splines:
# Con 4 nodos:
library("splines")
modelosplin1<- lm(y ~ bs(x, knots = nodos1, degree = 3),data=data.frame(x))
plot(x,y,ylab="Y")
points(x,fitted(modelosplin1),col="red",type="l",lwd=2)

# Con 12 nodos:
modelosplin2<- lm(y ~ bs(x, knots = nodos2, degree = 3),data=data.frame(x))
points(x,fitted(modelosplin2),col="green",type="l",lwd=2)
```

Código para el ajuste de los modelos de la Figura 2.6:

```
# Cálculo de la matriz de la base y la matriz de penalizaciones
nodos=seq(-4, 4, length.out = 48) #vector de nodos del intervalo [-4,4]
D<-diff(diag(k),differences=2) #matriz de diferencias de orden 2
P<-t(D)%*%D #matriz de penalizaciones
library("splines")
X=bs(x, df = NULL, knots = nodos, degree = 3, intercept = FALSE,  Boundary.knots = range(x))
 #matriz de la base de B-splines


```

```
# Ajuste del modelo para el parámetro de suavizado 0.001
lambda1=0.001 #parametro de suavizado
H1=X%*%solve(t(X)%*%X+lambda1*P)%*%t(X) #matriz hat
betagorro1=solve(t(X)%*%X+lambda1*P)%*%t(X)%*%y #estimación del vector beta
ygorro1=H1%*%y #valores ajustados
plot(x,y,ylab="Y")
points(x, ygorro1, type="l", col="red",lwd=2)

# Ajuste del modelo para el parámetro de suavizado 10
lambda2=10
H2=X%*%solve(t(X)%*%X+lambda2*P)%*%t(X)
betagorro2=solve(t(X)%*%X+lambda2*P)%*%t(X)%*%y%*%y
ygorro2=H2%*%y
plot(x,y,ylab="Y")
points(x, ygorro2, type="l", col="green",lwd=2)
```

Código para el ajuste de los modelos de la Figura 2.8 y el cálculo de los valores del Cuadro 2.1:

```
#Ajuste de los modelos
#modelo gam
mod_gam <- gam(y ~ s(x,k=-1,bs="ps"), family = binomial(link = logit),method="GCV.Cp")
summary(mod_gam)
plot(mod_gam)

#modelo glm
mod_glm <- glm(y ~ x, family=binomial(link="logit"))
summary(mod_glm)

#modelo generalizado linealizable orden 6
mod_glm_pol6<- glm(y ~ poly(x, 6), family=binomial(link="logit"))
summary(mod_glm_pol6)

#modelo generalizado linealizable orden 3
mod_glm_pol3<- glm(y ~ poly(x, 3), family=binomial(link="logit"))
summary(mod_glm_pol3)

```

```r
#modelo generalizado linealizable orden 12
mod_glm_pol12<- glm(y ~ poly(x, 12), family=binomial(link="logit"))
summary(mod_glm_pol12)

#Deviances explicadas
# Modelo glm
mod_glm_null <- glm(y ~ 1, family=binomial(link="logit")) # Modelo nulo
mod_glm_devnull <- deviance(mod_glm_null) # Devianza nula
mod_glm_dev_exp <- (mod_glm_devnull - deviance(mod_glm)) / mod_glm_devnull
# Deviance explicada

# Modelo generalizado linealizable polinomico de orden 6
mod_glm_pol6_null <- glm(y ~ 1, family=binomial(link="logit"))
mod_glm_pol6_devnull <- deviance(mod_glm_pol6_null)
mod_glm_pol6_dev_exp <-
(mod_glm_pol6_devnull - deviance(mod_glm_pol6)) / mod_glm_pol6_devnull

# Modelo generalizado linealizable polinomico de orden 3
mod_glm_pol3_null <- glm(y ~ 1, family=binomial(link="logit"))
mod_glm_pol3_devnull <- deviance(mod_glm_pol3_null)
mod_glm_pol3_dev_exp <-
(mod_glm_pol3_devnull - deviance(mod_glm_pol3)) / mod_glm_pol3_devnull

# Modelo generalizado linealizable polinomico de orden 12
mod_glm_pol12_null <- glm(y ~ 1, family=binomial(link="logit"))
mod_glm_pol12_devnull <- deviance(mod_glm_pol12_null)
mod_glm_pol12_dev_exp <-
(mod_glm_pol12_devnull - deviance(mod_glm_pol12)) / mod_glm_pol12_devnull

#Modelo gam
summary(mod_gam)$dev.expl

#Representación gráfica
plot(x,y, ylab="Y")
lines(x, fitted(mod_glm_pol3), type = "l", col = "pink", lwd = 1.5)
lines(x, fitted(mod_glm_pol12), type = "l", col = "green", lwd = 1.5)
lines(x, fitted(mod_glm), type = "l", col = "yellow", lwd = 1.5)

```

```
57 plot(x,y, ylab="Y")
58 lines(x, fitted(mod_gam), type = "l", col = "blue", lwd = 1.5)
59 lines(x, fitted(mod_glm_pol6), type = "l", col = "red", lwd = 1.5)
```

Bibliografía

Eilers, P. H. C., & Marx, B. D. (1996). Flexible Smoothing with B-splines and Penalties. *Statistical Science*, *11*(2), 89-121. https://doi.org/10.1214/ss/1038425655

Eilers, P. H. C., Marx, B. D., & Durbán, M. (2015). Twenty years of P-splines. *SORT-Statistics and Operations Research Transactions*, *39*(2), 149-186. https://raco.cat/index.php/SORT/article/view/302258

Faraway, J. J. (2004). *Linear models with R.* Chapman & Hall/CRC.

Faraway, J. J. (2006). *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models.* Chapman & Hall/CRC.

Green, P. J., & Silverman, B. W. (1993). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach.* Springer.

Perperoglou, A., Sauerbrei, W., Abrahamowicz, M., & Schmid, M. (2019). A review of spline function procedures in R. *BMC Medical Research Methodology*, *19*(1), 46. https://doi.org/10.1186/s12874-019-0666-3

Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993). Nuclear Feature Extraction for Breast Tumor Diagnosis. *Biomedical Image Processing and Biomedical Visualization, IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology.* https://minds.wisconsin.edu/bitstream/handle/1793/59692/TR1131.pdf?sequence=1

Wolberg, W., Mangasarian, O., Street, N., & Street, W. (1995). Breast Cancer Wisconsin (Diagnostic). https://doi.org/10.24432/C5DW2B

Wood, S. N. (2017). *Generalized Additive Models: An Introduction with R.* CRC Press.

Wood, S. N. (2023). mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation [R package version 1.9-1]. https://cran.r-project.org/web/packages/mgcv/mgcv.pdf